

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA



Desenvolvimento de um Sistema de Monitorização Remota do Consumo de Energia Elétrica e da Presença Humana em Edifícios

Ricardo Filipe Querido Henriques

Dissertação

Mestrado Integrado em Engenharia da Energia e do Ambiente

2012

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA



Desenvolvimento de um Sistema de Monitorização Remota do Consumo de Energia Elétrica e da Presença Humana em Edifícios

Ricardo Filipe Querido Henriques

Dissertação orientada pelos

Prof. Doutor Guilherme Carrilho da Graça (FCUL)

Prof. Doutor Jorge Maia Alves (FCUL)

Mestrado Integrado em Engenharia da Energia e do Ambiente

2012

Abstract

The actual world energy scenario, either economic or environmental, has been changing the way as the consumers face the use of resources. As a consequence people are getting aware of managing the energy consumption in buildings. In the case of electric energy the typical consumption control is done with low detail, thus it is difficult to establish cause-effect relationship between values of energy consumed and its origin.

The energy monitoring systems are presented as a solution capable to reveal to consumers the real consumption evolution, making possible to identify inefficient aspects and energy reduction measures. On the other hand, simultaneous monitoring with other parameters such as human presence in buildings, allows a better evaluation of occupation and consumption linkage.

A wide variety of electronic devices are available in the market at low cost and high quality level enabling the development of this monitoring systems, which can be a good alternative to the existent commercial equipment. Moreover, many of this devices work through open source principle, existing a lot of information to access and use freely.

The purpose of this thesis is to develop and test a system capable to monitor the electrical energy consumption and detect human presence in buildings. The system includes wireless communication, development of one computer interface to observe the real time data, storage and sharing the data collected on internet.

Keywords: electric energy consumption, monitoring system, energy management, energy efficiency.

Resumo

O atual cenário energético mundial, quer ao nível económico quer ambiental, tem alterado a forma como o consumidor encara a utilização dos recursos. Como consequência a população está mais sensibilizada para a gestão da energia utilizada nos edifícios convencionais. No caso da energia elétrica, o controlo típico dos consumos efetuados tem muito pouco detalhe, sendo difícil estabelecer relações de causa efeito entre os valores de energia consumida e a própria origem desse consumo.

No caso particular da eletricidade, os sistemas de monitorização energética apresentam-se como a solução capaz de revelar ao utilizador dos edifícios a real evolução dos próprios consumos, passando a ser possível identificar eventuais pontos de ineficiência e potenciais medidas de redução do consumo elétrico. Por outro lado, a monitorização em simultâneo com outros parâmetros, como por exemplo a presença humana em edifícios, permite uma melhor avaliação da relação existente entre ocupação e consumo.

A grande variedade de dispositivos eletrónicos disponíveis no mercado a baixo custo e com níveis de qualidade exigida, permitem o desenvolvimento deste tipo de sistemas de monitorização, podendo ser uma boa alternativa aos equipamentos comerciais existentes. Além disso, muitos destes dispositivos funcionam segundo o princípio do *open source*, sendo possível utilizar e aceder a muita informação de forma livre.

O propósito desta tese visa desenvolver e testar um sistema capaz de monitorizar o consumo de energia elétrica, bem como detetar a presença humana em edifícios. O sistema inclui comunicação sem fios, desenvolvimento de uma interface no computador para visualização dos dados em tempo real, armazenamento e partilha na internet dos dados recolhidos.

Palavras-chave: consumo de energia elétrica, sistema de monitorização, gestão de energia, eficiência energética.

Índice

Abstract	5
Keywords.	5
Resumo.....	7
Palavras-chave.....	7
Índice.....	9
1. Introdução.....	1
2. Consumo de Energia Elétrica em Edifícios.....	3
2.1 Conceitos Fundamentais.....	3
2.2 Tipos de Cargas Elétricas	5
2.3 Tipos de Instalações Elétricas	5
3. Sistemas de Monitorização de Energia Elétrica	6
3.1 Sistemas de Monitorização para o Setor Residencial	6
3.2 Sistemas de Monitorização para o Setor dos Serviços e da Indústria.....	7
3.3 Sistemas de Monitorização DIY	8
4. Sistema de Monitorização Proposto	9
4.1 Hardware	9
4.1.1 Arduino.....	9
4.1.2 Wireless SD Shield.....	11
4.1.3 Transformador de Corrente	11
4.1.4 Sensor de Infravermelhos Passivo.....	13
4.1.5 Sensor de Ultrassons	14
4.1.6 Relógio de Tempo Real	15
4.2 Software	16
4.2.1 Processing.....	16
4.2.2 Arduino IDE/Wiring.....	17
4.3 Comunicação.....	17
4.3.1 USB	17
4.3.2 I ² C.....	17

4.3.3	SPI	18
4.3.4	A Tecnologia ZigBee	18
4.4	Internet das Coisas (IoT-Internet of Things)	23
4.5	Montagem do Sistema	24
4.5.1	Hardware	24
4.5.2	Código Arduino IDE	27
4.5.3	Código Processing	28
4.5.4	Interface Cosm	30
4.6	Calibração do Sistema	30
4.6.1	Curvas de corrente medidas pelo TC.....	30
4.6.2	Tensão produzida pelo TC com resistência de carga	32
4.6.3	Verificação dos resultados produzidos pelos sensores IVP e ultrassons	33
4.6.4	Calibração da Corrente Eficaz.....	34
4.6.5	Teste ao Sistema Proposto sem Zigbee	35
4.6.6	Teste ao protótipo com utilização do Zigbee.....	36
5.	Aplicação do Sistema Proposto	37
5.1	Medições e Resultados	38
5.1.1	Corrente Eficaz.....	38
5.1.2	Tensão Eficaz	42
5.1.3	Fator Potência.....	42
5.1.4	Potência Aparente.....	43
5.1.5	Potência Ativa	44
5.1.6	Energia Elétrica	44
5.1.7	Presença.....	46
6.	Conclusões	47
7.	Referências	49
	Anexo A: Preços do Material	51
	Anexo B: Código de Programação do Arduino.....	52
	Anexo C: Código de Programação do Processing.....	56

Lista de Figuras

Fig. 1 – iMeter da ISA (Fonte: Fabricante)	6
Fig. 2 - CA 8332 da Chauvin Arnoux (Fonte: Fabricante).....	7
Fig. 3- Projeto Open Energy Monitor	8
Fig. 4 - Estrutura do Sistema Desenvolvido	9
Fig. 5 – Esquema Arduino Uno R3	10
Fig. 6 - Arduino Wireless SD Shield (Fonte: Inmotion)	11
Fig. 7- Circuito Elétrico de um Transformador (Fonte: Wikipedia)	12
Fig. 8 - CR Magnetics CR3110-3000 (Fonte: DigiKey)	12
Fig. 9- Hanse Electronics SE-10 (Fonte: Sparkfun).....	14
Fig. 10 - Devantech SRF02 (Fonte: Solarbotics)	15
Fig. 11- Maxim DS1307.....	16
Fig. 12 - Arquitetura do Protocolo ZigBee (ZigBee Standards Organization, 2006).....	19
Fig. 13 - Topologias de rede para a norma IEEE. 802.15.4 (SAFARIC, et al., 2006)	20
Fig. 14 – Digi International Xbee (Fonte: Sparkfun)	21
Fig. 15 - Configurações dos módulos utilizados, à esquerda a do Router e à direita a do Coordenador	22
Fig. 16 - Representação da estrutura da Internet das Coisas (Fonte: Doukas, 2012)	23
Fig. 17- Esquema eletrônico global do projeto	24
Fig. 18 - Fluxograma do <i>sketch</i> utilizado na programação do Arduino	27
Fig. 19 - Fluxograma do <i>sketch</i> utilizado na programação do Processing	28
Fig. 20 - Interface de monitorização do sistema desenvolvido	29
Fig. 21 - Interface produzido pelo Cosm.....	30
Fig. 22- Gráficos representativos da corrente consumida pelo portátil, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita).....	31
Fig. 23- Gráficos representativos da corrente consumida pelo ferro de engomar, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita).....	31
Fig. 24 - Gráficos representativos da corrente consumida pelo candeeiro, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita).....	32
Fig. 25 - Gráficos representativos da corrente consumida pelo portátil mais o candeeiro, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita)	32
Fig. 26- Gráfico da tensão produzida para vários valores de corrente no primário ($R_{carga}=180\Omega$).....	33
Fig. 27 - Gráfico da variação do consumo de energia elétrica e presença no gabinete	35
Fig. 28 - Gráfico da variação do consumo de energia elétrica e presença no laboratório	36
Fig. 29 - Montagem final dos dois sistemas a instalar no bar do C1	37
Fig. 30 - Gráfico de comparação da corrente eficaz para o período total de monitorização (resultados do protótipo a verde e resultados do CA8332 a azul)	38

Fig. 31 - Gráfico de comparação da corrente eficaz para o horário com ocupação num dia útil (resultados do protótipo a verde e resultados do CA8332 a azul)	39
Fig. 32 - Gráfico de comparação da corrente eficaz para o horário sem ocupação num dia útil (resultados do protótipo a verde e resultados do CA8332 a azul)	40
Fig. 33 - Gráfico de comparação da corrente eficaz num dia de fim de semana (resultados do protótipo a verde e resultados do CA8332 a azul)	41
Fig. 34 - Gráfico de comparação da tensão eficaz (resultados do protótipo a verde e resultados do CA8332 a azul).....	42
Fig. 35- Gráfico de comparação do fator potência (resultados do protótipo a verde e resultados do CA8332 a azul).....	43
Fig. 36 - Gráfico de comparação da potência aparente consumida (resultados do protótipo a verde e resultados do CA8332 a azul)	43
Fig. 37 - Gráfico de comparação da potência ativa consumida.....	44
Fig. 38 - Gráfico de comparação do consumo de energia elétrica (resultados do protótipo a verde e resultados do CA8332 a azul)	45
Fig. 39- Gráfico indicativo da presença humana no bar.....	46

Lista de Tabelas

Tabela 1 - Parâmetros monitorizados para vários equipamentos analisados.....	34
Tabela 2 - Exemplo dos erros presentes nos valores obtidos utilizando o Xbee	36
Tabela 3 – Características técnicas de cada sistema.....	37
Tabela 4 – Parâmetros medidos por cada sistema	37
Tabela 5 - Horário de funcionamento do bar do C1	38
Tabela 6 - Comparação dos valores de energia consumida, custo da energia e CO ₂ emitido	45
Tabela 7 - Valores da primeira e última deteção para cada dia da monitorização	46

Lista de Siglas

AC	Alternating Current
ADC	Analog-to-Digital Converter
API	Application Programming Interface
AVAC	Aquecimento, Ventilação e Ar Condicionado
AT	Transparent Mode
CA	Chauvin Arnoux
CSV	Comma Separated Value
DC	Direct Current
E/S	Entrada/Saída
EEPROM	Electrically Erasable Programmable Read-Only Memory
GND	Massa/Terra
I ² C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IEE	Indicador de Eficiência Energética
IoT	Internet of Things
IVP	Infravermelhos Passivo
MISO	Master In Slave Out
MOSI	Master Out Slave In
PWM	Pulse With Modulation
RCCTE	Regulamento das Características de Comportamento Térmico de Edifícios
RMS	Root Mean Square
RSECE	Regulamento dos Sistemas Energéticos de Climatização em Edifícios
RTC	Real Time Clock
RSSF	Rede de Sensores Sem Fios
SCL	Serial Clock Line
SD	Secure-Digital
SDA	Serial Data Line
SEE	Sistemas de Energia Elétrica
SME	Sistema de Monitorização de Energia
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SS	Slave Select
TC	Transformador de Corrente
TT	Transformador de Tensão
USB	Universal Serial Bus
WPAN	Wireless Personal Area Network
V _{cc}	Tensão de Alimentação em Corrente Contínua

1. Introdução

A eletricidade é no século XXI um bem, considerado por grande parte da população mundial, equiparável à água e alimentos. É de tal forma essencial à sociedade atual que seria difícil imaginar como seria o Mundo, caso não existisse energia elétrica. A qualidade de vida nos países desenvolvidos requer um elevado consumo energético, o que se traduz num grande impacto ambiental, que há muito ultrapassou a fronteira da sustentabilidade. Uma das principais causas para o aumento significativo do consumo energético, encontra-se nas necessidades energéticas dos edifícios construídos ao longo das últimas décadas, ao nível da sua climatização, iluminação e consumo por parte de outros equipamentos. Os cenários energéticos, apesar de não serem claros ao nível concreto dos edifícios, apontam para que estes representem entre 20 a 40% do consumo energético total nos países desenvolvidos (Lombard, et al., 2007). Desta percentagem total, o consumo elétrico em edifícios residenciais e de serviços, apresentam um valor de 56,2% (IEA, 2009)¹. Tendo em conta os dados referidos, o consumo elétrico, revela ter um papel bastante importante no consumo global dos edifícios.

O ciclo da energia elétrica é composto por várias etapas, que incluem a sua produção e transporte até ao utilizador final. O transporte é dividido numa primeira fase na rede de transmissão e numa segunda na rede de distribuição. O último processo da entrega da energia elétrica é a sua entrega ao consumidor, sendo este passo realizado por empresas de comercialização de eletricidade. Desde o início da sua comercialização, a forma de quantificar o consumo de eletricidade tem sido efetuada a partir de contadores, colocados à entrada das casas ou em edifícios. Desta forma é possível a um responsável da empresa de comercialização da energia, fazer uma contagem da mesma, sendo o valor apresentado numa fatura mensal/bimestral. No caso português, nos contratos estabelecidos para baixa e média tensão, o nível de detalhe das faturas é bastante reduzido, sendo apresentados os valores de energia elétrica consumida, além do preço a pagar pela entidade (EDP Energias de Portugal, 2012). Para contratos de média tensão são também taxados os consumos de energia reativa. Embora seja apresentado um valor de consumo para o horário de vazio e fora de vazio, é extramente difícil ao consumidor identificar e caracterizar a variação do seu consumo temporalmente.

Atendendo ao atual cenário energético e ambiental do mundo, caracterizado por um aumento do preço da energia e pelo problema das alterações climáticas, o facto da grande maioria dos edifícios não ser continuamente monitorizado ao nível do consumo elétrico, resulta numa menor consciencialização da população para a forma como gasta os seus recursos. É pois, de todo o interesse, o desenvolvimento de sistemas versáteis de monitorização de consumos de energia.

Desta forma, a monitorização do consumo de energia elétrica enquadra-se como um dos procedimentos da Gestão de Energia (M&T-Energy Monitoring and Targeting), sendo apresentada como um dos principais meios para estudar potenciais medidas de eficiência energética aplicadas a um edifício, com vista à redução do consumo. Partindo do princípio de que “é impossível reduzir algo que não se conhece”, um estudo deste género permite observar em detalhe uma série de aspetos, que, tomando como exemplo as faturas de eletricidade atuais, não poderiam ser tidos em consideração (Bureau of Energy Efficiency). Um sistema capaz de desempenhar as funções de gestor de energia, deve então incluir três elementos chave (Dent, 2010):

- dispositivo de monitorização ou medição;
- armazenamento de dados;
- análise de dados.

Em relação ao sistema de monitorização de energia (SME), revela-se de extrema importância para a concretização da medição, pois é ele que vai permitir registar os valores de consumo reais. Para a sua seleção é determinante um conhecimento atempado das tecnologias de monitorização existentes, e dos

¹ Cenário da IEA analisado inclui agricultura e outros setores não especificados

condicionalismos associados à medição a realizar, por exemplo em termos do espaço físico (ex.: espaço existente no quadro elétrico, grossura dos cabos elétricos).

O armazenamento de dados é por sua vez um elemento fundamental em todo este processo, porque permite agregar de uma forma organizada os dados. Como este tipo de monitorizações envolve guardar dados durante séries de tempo que podem ir de dias a meses, é necessário ter uma certa memória interna ou externa disponível.

A análise de dados, apesar de não ser um dos objetivos da tese desenvolvida, revela-se um dos elementos preponderantes para detetar eventuais erros do sistema de monitorização, validando ou não o seu funcionamento. Normalmente nesta fase são utilizados gráficos para uma melhor perceção, do perfil energético do espaço ou edifício, e da variação do consumo a eventuais medidas de eficiência energética implementadas.

Através do conjunto de elementos do gestor de energia, passa a ser possível um registo do consumo energético, correspondente a uma escala temporal previamente definida, que possibilitará a identificação de aumentos e diminuições do consumo energético. Assim, tendo conhecimento das necessidades energéticas do edifício em questão, é possível obter uma relação de causa efeito entre o cenário observado e o comportamento dos utilizadores e equipamentos do edifício, podendo ser detetadas medidas capazes de reduzir o atual cenário de consumo energético, ou de problemas de funcionamento de equipamentos utilizados que resultam em perdas de energia. Com a implementação das medidas, e nova monitorização ou estimação da evolução do consumo, é possível comparar e calcular a percentagem de redução do mesmo consumo.

Um exemplo deste novo paradigma no setor energético, em relação à alteração da forma como é feita a contagem energética, quer seja elétrica, de gás, ou calor, é o incentivo por parte da União Europeia à instalação de novos aparelhos de contagem eletrónica, capazes de mostrar o consumo real de energia do consumidor final além de informações relacionadas com o real período de utilização (Comunidade Europeia, 2006). Alguns dos aspetos-chave destes sistemas segundo a mesma diretiva, passam pelo custo competitivo, gestão à distância e dados detalhados, sendo estes fatores considerados no sistema desenvolvido.

A grande maioria dos utilizadores dos edifícios não tem consciência de que forma as suas ações podem afetar a fatura energética mensal. No entanto, existem vários estudos que demonstram que após a introdução de sistemas de monitorização tanto em habitações como outro género de edifícios, os mesmo utilizadores além de perceberem a evolução do seu consumo, alteram os seu hábitos de consumo de energia elétrica. Tendo em conta o tipo de *feedback* devolvido pelos SME, e o tipo de interface utilizado para visualizar os dados obtidos, é possível obter valores de redução de consumo, que no caso da energia elétrica podem variar entre os 5 e os 15% (Darby, 2006). Outra das vantagens, da utilização dos SME, passa também pela sensibilização dos consumidores relativamente ao período de tempo em que utilizam determinados equipamentos, tendo em conta as várias tarifas existentes.

Por outro lado, no caso de um espaço como o analisado na dissertação, é extremamente relevante a medição de uma série de outros parâmetros além do consumo energético. Fatores como as condições de conforto interior (ex.: temperatura, dióxido de carbono, luminosidade e humidade relativa) e presença de pessoas num edifício, são preponderantes num estudo deste tipo, pois quando intersetados com as variações do consumo energético possibilitam uma série de correlações, permitindo a identificação de eventuais pontos de ineficiência na utilização do mesmo espaço. Por exemplo os consumos no horário sem ocupação são passíveis de serem reconhecidos através da colocação de sensores capazes de detetar a presença de pessoas, ou a identificação de equipamentos deixados em *standby*. Por outro lado uma das principais causas para o aumento do consumo energético de um edifício neste momento é a climatização do mesmo por sistemas AVAC (Aquecimento, Ventilação e Ar Condicionado). A medição das condições de conforto e variação do consumo energético pode consentir a identificação de períodos de tempo, em que o espaço não necessita de climatização, mas apesar disso pelo consumo energético se observa que o mesmo é climatizado. Para evitar este desperdício de energia e consequente redução de consumo, a automação do próprio edifício surge como uma solução complementar à monitorização.

2. Consumo de Energia Elétrica em Edifícios

Para que o sistema de monitorização possa ser desenvolvido, importa saber de que forma o consumo de energia elétrica em edifícios se manifesta, destacando os vários conceitos essenciais à sua compreensão:

- corrente alternada;
- tensão;
- potência ativa, reativa e complexa;
- fator potência;
- energia.

2.1 Conceitos Fundamentais

Os sistemas de energia elétrica (SEE) convencionais são caracterizados por operarem em corrente alternada, correspondendo esta ao movimento de cargas elétricas que variam periodicamente no seu sentido. Como resultado da corrente alternada é possível observar uma onda sinusoidal. Outro dos termos indispensáveis na área da eletricidade é a tensão elétrica, que indica o trabalho necessário para movimentar uma carga contra um campo elétrico. No caso dos SEE os valores de tensão vão variando à medida que se vai avançando, desde o ponto em que a eletricidade é produzida até ao ponto em que é entregue. Por exemplo, no que diz respeito ao valor de tensão nominal encontrado num edifício em Portugal, ele é aproximadamente de 230V para uma instalação monofásica.

Através do produto do valor da corrente e tensão, num determinado ponto, é possível obter a potência instantânea, ou seja a energia consumida por segundo. A variação da corrente $i(t)$ e da tensão $u(t)$ origina curvas sinusoidais, onde a potência pode ser calculada por duas fórmulas:

$$P = \frac{1}{T} \int_0^T u(t) \times i(t) dt \equiv U \times I \times \cos(\varphi) \quad (1)$$

Na primeira igualdade da equação 1 é calculado um valor médio da potência instantânea, P , através da integração da tensão multiplicada à corrente, durante um certo intervalo de tempo (Atmel, 2007). Esta potência denomina-se de ativa e representa a potência consumida para produção de trabalho útil, tendo como unidades o Watt. Em relação à segunda igualdade da equação 1 representa o cálculo da mesma potência instantânea, mas desta feita através dos valores eficazes (RMS-Root Mean Square) de corrente I , e tensão U , multiplicados ao fator potência, $\cos(\varphi)$. Tal como é indicado, as fórmulas equiparam-se, permitindo chegar ao mesmo valor da potência ativa.

Importa explicar que o valor eficaz parte da utilização de uma medida de estatística que permite o cálculo da magnitude de uma quantidade variável, neste caso a tensão e a corrente. Tal como o seu nome em inglês indica, é uma raiz quadrada da média dos quadrados dos valores.

Admitindo que tanto a tensão como a corrente seriam medidas pelo sistema desenvolvido, a potência em termos de tempo discreto seria:

$$P = \frac{1}{N} \int_0^{N-1} u(n) \times i(n) \quad (2)$$

A utilização do tempo discreto é imprescindível, pois permite uma aproximação bastante aceitável ao que seriam valores de corrente e tensão em tempo contínuo. Assim dividindo um ciclo em várias amostras discretas, num total de N amostras, permite o cálculo de um valor médio de potência, sendo válido para curvas sinusoidais ou de forma distorcida.

Como já foi referido nas fórmulas anteriores são utilizados valores eficazes de tensão e corrente. Partindo deste método de estatística, tem-se que, a corrente eficaz é calculada através de:

$$I = \sqrt{\frac{\sum_{n=0}^{N-1} i^2(n)}{N}} \quad (3)$$

O cálculo da tensão eficaz é efetuado a partir da equação 3, com exceção da utilização dos valores discretos de tensão.

No caso do projeto desenvolvido não foi utilizado um sensor capaz de medir os valores de tensão, tendo sido tomado como constante o valor de 230V para a tensão eficaz.

Além da existência da potência ativa no fluxo energético são definidas outras duas formas de potência, a reativa (Var) e a complexa (VA). Entende-se pela primeira, a potência que não transfere energia, ou seja, apenas é utilizada para apoio na transferência de potência ativa na rede elétrica. O valor médio da potência reativa é nulo, e resulta da variação da energia magnética ou elétrica armazenada em elementos capacitivos ou indutivos, traduzindo-se numa inversão do sentido do fluxo energético (Paiva, 2007). Devido ao facto de não transferir energia é representada através de um número imaginário.

No que diz respeito à potência complexa, não é mais do que a soma vetorial dos valores das restantes potências. Normalmente o valor medido neste caso é a potência aparente, que resulta da magnitude do vetor produzido pela soma da potência ativa com a reativa. A potência aparente pode ser calculada através do produto entre os valores eficazes de tensão e corrente:

$$|S| = U \times I = \sqrt{\frac{\sum_{n=0}^{N-1} i^2(n)}{N}} \times \sqrt{\frac{\sum_{n=0}^{N-1} u^2(n)}{N}} \quad (4)$$

Como foi mencionado, para se obter a potência ativa é necessário o valor do fator de potência. Este fator traduz-se como a razão entre a potência ativa e a potência aparente, sendo o ângulo da fase, φ , o ângulo entre a tensão e a corrente.

No sistema proposto, caso fosse medida a tensão, seria possível o cálculo da potência ativa e da potência aparente, contudo o facto de se assumir a tensão eficaz como um valor constante, torna impossível o cálculo do fator de potência para cada instante. Como o cálculo da potência ativa também está condicionado pela existência de valores reais de tensão eficaz, a solução encontrada passa por definir também o fator potência como um valor constante. No caso da tese foi selecionado o valor de 0,7 a partir do fator de potência observado para uma série de equipamentos, tendo-se revelado uma boa escolha como comprovarão os resultados finais obtidos. Deste modo partindo da equação 5, passa a ser possível o cálculo da potência ativa.

$$P = |S| \times \cos(\varphi) = \left(\sqrt{\frac{\sum_{n=0}^{N-1} i^2(n)}{N}} \times 230 \right) \times 0,7 \quad (5)$$

O conhecimento do valor da potência ativa, torna possível o cálculo da energia elétrica consumida num dado intervalo de tempo através da sua integração para o período de tempo analisado:

$$E = \int_0^T P(t) \times dt \quad (6)$$

2.2 Tipos de Cargas Elétricas

Anteriormente foram referidas três formas de potência que podem ser observadas num sistema elétrico. A variação dos seus valores está relacionada com o tipo de interação entre os equipamentos e o sistema elétrico, sendo possível observar distintos comportamentos. Entre os conceitos de cargas mais conhecidos encontram-se as cargas resistivas e reativas, indicadas para a avaliação do sentido do fluxo energético, e lineares e não lineares, para avaliação das curvas relativamente à proximidade a uma senoide.

Os equipamentos do tipo carga resistiva são definidos como capazes de utilizar toda a energia que lhes é fornecida, dissipando-a sob a forma de calor, sendo o circuito elétrico representado apenas pela existência de resistências elétricas (Lei de Ohm). Como neste caso a polaridade da corrente e da tensão variam ao mesmo tempo (estão em fase) a potência é sempre positiva, o que permite concluir que o sentido do fluxo de energia nunca se inverte. Desta forma o fator potência tem sempre o valor de 1, sendo a potência reativa nula e a potência ativa igual à aparente. Como exemplos de equipamentos deste género têm-se lâmpadas incandescentes, ferros de engomar e grelhadores elétricos.

Por outro lado, equipamentos do tipo carga reativa comportam-se de forma diferente, sendo que neste caso a corrente e a tensão já não estão em fase, o que se traduz em períodos de tempo em que o sentido do fluxo energético é negativo. Isto implica que a potência instantânea tenha períodos em que é positiva e outros em que é negativa, correspondendo à criação de um valor de potência reativa. Em relação ao fator potência nesta situação é inferior a 1, variando o valor de acordo com as características dos equipamentos conforme se aproximem mais do puramente reativo (valor 0) ou do puramente resistivo (valor 1). Para este caso, encontram-se nos equipamentos componentes como condensadores ou indutores. Os frigoríficos e máquinas de lavar são exemplos de equipamentos de carga reativa. Relativamente à linearidade da curva de corrente, caso tenha a forma de uma senoide é uma carga linear, indicando que em qualquer momento a corrente é proporcional à tensão. Exemplos de equipamentos com estas características são os puramente resistivos, como o ferro de engomar, ou a lâmpada incandescente.

Ao invés, nas cargas não lineares a corrente não é proporcional à tensão. Este tipo de cargas pode ser causado, por exemplo, por retificadores que provocam distorções na corrente. A maioria dos aparelhos eletrónicos, tais como computadores portáteis, frigoríficos, televisões, são cargas deste tipo.

Na secção de da calibração do sensor de corrente são estudados alguns equipamentos relativamente à forma da curva de corrente e ao seu tipo de carga.

2.3 Tipos de Instalações Elétricas

Para a escolha do tipo de sistema de monitorização a desenvolver é importante conhecer os tipos de instalações elétricas existentes em termos de quantidades de fases: monofásica ou trifásica. No caso da monofásica, maioritariamente presente no setor residencial, do quadro elétrico de cada habitação saem três cabos: a fase, o neutro e a terra. Por sua vez numa instalação trifásica saem cinco cabos, sendo três deles para cada uma das fases, e os restantes dois para a terra e para o neutro. Não obstante a simplicidade da função de cada cabo, o seu conhecimento é de extrema importância aquando da instalação do sistema desenvolvido nos quadros elétricos. Assim no caso das fases, é neste cabo que passa a corrente que alimenta os equipamentos de um espaço/edifício, enquanto no neutro passa a corrente de retorno. No caso dos quadros monofásicos o valor da corrente que passa na fase é aproximadamente igual ao do neutro, o que permite que o sensor de corrente seja instalado em ambos. Por outro lado nos quadros elétricos trifásicos o valor da corrente no neutro pode não corresponde à soma das correntes de todas as fases.

3. Sistemas de Monitorização de Energia Elétrica

Devido à importância que os aparelhos de monitorização têm ganho nos últimos anos, têm surgido no mercado uma série de soluções, tanto para o setor doméstico como para o setor dos edifícios de serviços ou fábricas, variando as capacidades e características dos aparelhos para cada um destes setores. A complexidade, quantidade de parâmetros medidos e calculados, além do erro dos próprios valores, varia com o tipo de aparelho e medição que se está a efetuar. O princípio de funcionamento dos sistemas de monitorização de energia elétrica (SMEE) segue em termos gerais, as indicações de um gestor de energia, sendo constituído por:

- dispositivo capaz de adquirir dados, normalmente microcontrolador;
- transformadores/sensores de corrente (TC) e tensão (TT) capazes de transmitir sinais analógicos proporcionais aos valores de corrente e tensão medidos;
- interface para visualização de dados;
- possibilidade de armazenamento dos dados em cartão de memória, e/ou exportação por rede sem fios ou por fios.

De seguida são apresentados alguns exemplos de SMEE já existentes no mercado, ou desenvolvidos por pessoas particulares, com características de monitorização distintas.

3.1 Sistemas de Monitorização para o Setor Residencial

Em relação ao setor doméstico encontram-se muitas soluções capazes de oferecer o mesmo género de resultados, utilizando na maioria dos casos o mesmo princípio de funcionamento dos sistemas monofásicos. Uma das principais características dos sistemas atualmente comercializados é o facto de não incluírem a medição da tensão através de transformadores de tensão. Isto acontece, pois assume-se o valor da tensão da rede como constante, no caso português de 230V.



Fig. 1 – iMeter da ISA (Fonte: Fabricante)

Um exemplo de um medidor indicado para habitação é o iMeter, desenvolvido pela empresa ISA, que neste momento é vendido pela Portugal Telecom (Fig. 1). O conjunto é constituído por um sensor de corrente de núcleo aberto (mede até 160A de corrente), um mostrador digital para visualização dos consumos, e um iMeterBox que envia os dados para a rede de internet de casa (Portugal Telecom, 2012). Os valores de corrente registados pelo medidor são enviados através de um transmissor sem fios, para o mostrador digital e para o próprio iMeterBox, podendo também ser analisados numa interface existente na internet ou na própria televisão, os valores do consumo, custo e pegada de carbono da eletricidade consumida, para intervalos de tempo, diários, mensais ou anuais. Tal como foi referido nos conceitos básicos, este sistema tem a desvantagem de não permitir a medição do valor real de tensão, impossibilitando o cálculo do fator de potência, que por sua vez é necessário à obtenção do exato valor de potência ativa. Assim caso se estejam a medir consumos de equipamentos com características reativas ou indutivas os valores mostrados, podem não corresponder aos reais.

Além deste equipamento, a ISA desenvolveu recentemente o sistema Cloogy, com características bastante semelhantes, mas com a particularidade de incluir tomadas capazes de medirem o consumo individual realizado pelos equipamentos, permitindo observar a sua representatividade no consumo global. Tanto estas tomadas como os restantes dispositivos comunicam através de Zigbee, sendo que outra das particularidades das tomadas é a de se ligarem e desligarem remotamente.

Apesar de, segundo o fabricante do iMeter, este estar preparado para se ligar a mais dois sensores de corrente, e assim realizar medições em instalações trifásicas, é de esperar que o facto de não se medir a tensão real, se traduza em alguns erros nos resultados. Devido a estas imprecisões foram desenvolvidos aparelhos de medição próprios para monitorização de consumos de energia elétrica em instalações trifásicas.

3.2 Sistemas de Monitorização para o Setor dos Serviços e da Indústria

Aparelhos como o CA 8332 da Chauvin Arnoux (Fig. 2), que foi utilizado para efeitos de comparação de resultados na tese, são excelentes soluções para edifícios de serviços e industriais. Têm características técnicas que lhe permitem medir bastantes parâmetros, possibilitando uma análise mais exaustiva ao consumo de energia elétrico.



Fig. 2 - CA 8332 da Chauvin Arnoux (Fonte: Fabricante)

Este equipamento possui três sensores de corrente, três sensores de tensão para cada uma das fases e um sensor de tensão para medir a tensão no neutro (Chauvin Arnoux, 2004). O material utilizado é próprio para quadros elétricos que se anteveem com bastante maior dimensão que um monofásico, traduzindo-se numa maior segurança na própria medição. No caso particular desta marca, são disponibilizados uma série de sensores de corrente com distintas características técnicas, tanto ao nível do intervalo de correntes a medir, como diâmetro de abertura do núcleo. Em relação aos sensores de corrente eles permitem medir um máximo de 6500A enquanto os sensores de tensão podem medir tensões até 1000V.

Tendo os valores medidos por ambos os sensores, o algoritmo de cálculo utilizado pelo aparelho permite calcular os valores eficazes de tensão e corrente, os vários tipos de potência, a energia e o fator potência. De referir que os métodos de cálculo utilizados pelo algoritmo são semelhantes aos mencionados nos conceitos fundamentais, ou seja, iguais aos que vão ser implementados no sistema proposto. Através do uso de outras fórmulas o equipamento possibilita também o cálculo de uma série de outros parâmetros como o fator de distorção harmónico (*Total Harmonic Distortion-THD*) e a frequência.

Além dos sensores de medida, o equipamento incorpora um monitor LCD que disponibiliza os vários dados calculados tanto sobe a forma de gráficos como de valores, em tempo real. Este sistema tem a desvantagem de apenas guardar os dados numa memória interna e de os transmitir para um computador unicamente por cabo, tornando difícil o acompanhamento em tempo real dos valores medidos.

Para análise dos dados, o equipamento é acompanhado por um software próprio capaz de produzir gráficos para cada um dos parâmetros, além de guardar os vários dados em folhas de cálculo.

Este género de equipamentos devido a todas estas características possuem um preço mais elevado que os sistemas residenciais.

3.3 Sistemas de Monitorização DIY

Apesar da presença de todos estes sistemas no mercado atual, é de salientar a existência de alguns projetos que seguem a filosofia do “Faz Tu Mesmo” (DIY-*Do It Yourself*), sendo o resultado final igual ou melhor. Um dos melhores exemplos é o OpenEnergyMonitor (OEM), onde uma série de especialistas na área têm vindo a desenvolver vários sistemas de monitorização energética (consumo de energia elétrica, produção de energia elétrica por fotovoltaico, etc).

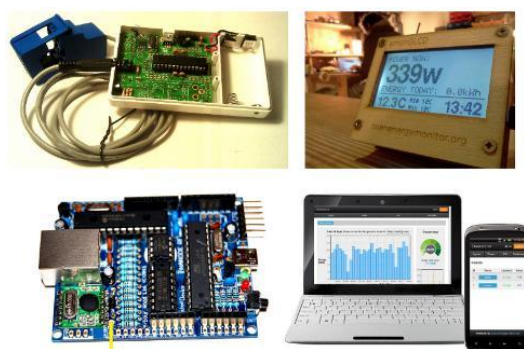


Fig. 3- Projeto Open Energy Monitor

Como se pode ver pela Fig. 3, no caso particular do OEM, o objetivo é em tudo semelhante ao pretendido na dissertação, sendo utilizado um microcontrolador semelhante ao Arduino, que comunica por rádio os dados medidos e calculados para um LCD e para uma interface construída na internet. Neste momento o sistema está direcionado para instalações monofásicas, pois apenas utiliza um sensor de corrente capaz de medir até 100A.

O sistema desenvolvido pelo OEM distingue-se do iMeter abordado anteriormente, por o algoritmo estar preparado a receber valores instantâneos de tensão fornecidos por um sensor a ligar. Em comparação com o CA 8332 tem a desvantagem de os sensores não permitirem alcançar valores tão elevados.

4. Sistema de Monitorização Proposto

Utilizando os princípios da Gestão de Energia chegou-se a um esquema final do que seria o sistema de monitorização a desenvolver. Este sistema envolve uma série de aspetos, como o hardware utilizado na construção do protótipo, o software utilizado na sua programação e na construção da interface, além das diferentes tecnologias de comunicação entre sensores/microcontrolador e microcontrolador/computador (ver Fig. 4).

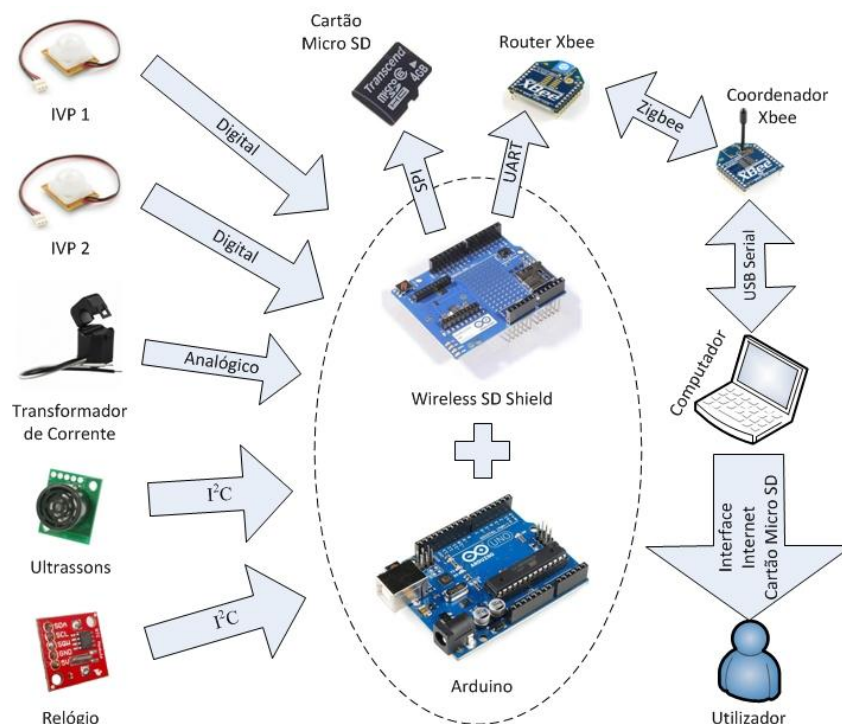


Fig. 4 - Estrutura do Sistema Desenvolvido

Para construção do protótipo foram definidos uma série de outros aspetos como por exemplo, o baixo custo que deveria ter o conjunto do equipamento a utilizar, bem como a versatilidade da sua utilização e o facto de poder ser escalável. Esta escalabilidade, é deveras importante, tendo em consideração, a possibilidade do sistema vir a servir de base aos SME a implementar no Campus da própria Universidade para a sua reabilitação energética (SESUL, 2011).

4.1 Hardware

Em relação ao hardware utilizado, é utilizado o microcontrolador Arduino, para a aquisição de dados, sendo estes obtidos a partir de um conjunto de sensores: transformador de corrente, sensor de infravermelhos passivo e sensor de ultrassons, além de um relógio para contagem do tempo real. De forma a permitir que se guardem os dados obtidos num cartão de memória micro SD e se comuniquem os mesmos dados por Zigbee para um computador, utiliza-se uma *shield*.

4.1.1 Arduino

O Arduino é uma plataforma de desenvolvimento eletrónico de *open source*, que funciona em torno de um microcontrolador ATMEAL AVR de 8 bits. A placa tem capacidade para importar e exportar conteúdo, sendo a linguagem de programação utilizada denominada como *Wiring* do Arduino ou Arduino IDE. O facto do projeto Arduino se basear todo ele na filosofia código aberto, foi extremamente importante no âmbito da tese, já que permitiu obter uma grande quantidade de hardware e códigos próprios para esta plataforma, reduzindo tempo e custos de toda a fase de estudo. (Arduino, 2012).

Apesar de existirem várias versões modelos do Arduino, todas têm como princípio comum a utilização do microcontrolador ATMEL AVR, variando este em termos das suas capacidades de processamento.

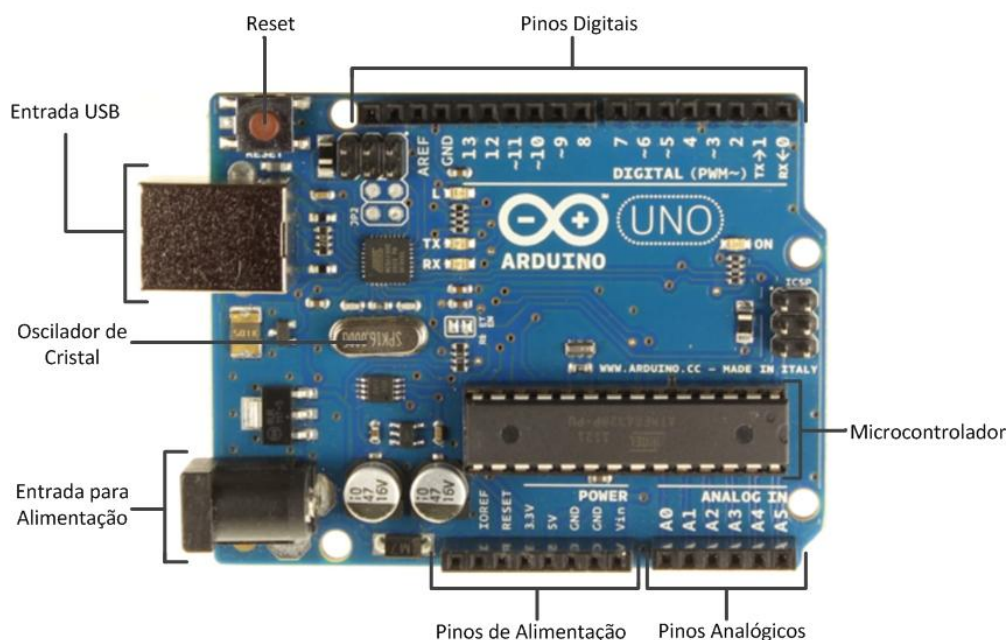


Fig. 5 – Esquema Arduino Uno R3

No âmbito da tese, a escolha recaiu no modelo Arduino Uno R3, por ser um dos modelos mais recentes e ter todas as características necessárias à realização das medições (Fig. 5).

A base para o funcionamento do Arduino Uno R3 é o seu microcontrolador ATmega328, que combina uma série de características como os três tipos de memórias utilizadas. Assim, no caso da memória flash existem 32 KB responsáveis pelo armazenamento dos programas carregados no Arduino a partir do Ambiente de Desenvolvimento, dos quais 0,5 KB são utilizados no *bootloader*. Relativamente à memória SRAM (*Static Random Access Memory*) utilizada na execução e alteração dos dados presentes na memória flash, e que se assemelha à memória RAM dos computadores normais, o ATmega328 possui 2 KB. Por sua vez este microcontrolador contém 1 KB de memória EEPROM (*Electrically Erasable Programmable Read-Only Memory*), que permitem que os dados não sejam apagados quando o Arduino não é alimentado eletricamente.

O Arduino é capaz de fornecer energia a partir dos seus pinos a uma tensão máxima de 5V ou 3,3V, e a uma corrente máxima de 50mA no pino de 3,3V e de 40mA nos pinos de E/S e de 5V, sendo assim possível alimentar os sensores e componentes eletrónicos a ele conectados. Em relação à alimentação da própria placa esta pode ser feita de duas formas distintas, através de ligação USB ou por fonte de alimentação periférica. Existem dois locais onde a fonte de alimentação periférica se pode ligar no Arduino: o *jack*, para ligação de transformadores AC/DC com entrada fêmea compatível, ou o pino Vin e *Ground* para ligação de uma bateria. Apesar de o Arduino estar apto a ser alimentado entre 6 a 20V, é recomendado pelo fabricante que seja alimentado por 7 a 12V.

Um dos aspetos essenciais à interação entre o Arduino e os sensores e componentes eletrónicos são os pinos de entrada e saída, que permitem ler e escrever dados. Assim existem na placa 14 pinos digitais, sendo que 6 deles permitem realizar PWM (*Pulse Width Modulation*).

Por outro lado o modelo Uno possui 6 pinos analógicos ligados a um conversor analógico/digital (ADC), que neste caso tem 10bits. Sabendo que o ADC tem 10 bits é possível calcular quantos níveis de tensão terá o mesmo, $2^{10}=1024$. Assim, como o mesmo ADC suporta um valor de tensão até um máximo de 5V, é possível calcular a resolução do ADC e das medições analógicas efetuadas:

$$Resolução_do_ADC = \frac{V_{máximo_Arduino}}{níveis_de_tensão} = 0,049V \quad (7)$$

Tal como qualquer outro microcontrolador, o Arduino possui um cristal com uma frequência de *clock* de 16 MHz, responsável pela velocidade de processamento das instruções realizadas.

Outra das características que torna o Arduino uma ferramenta tão utilizada em *physical computing*, é o facto de suportar várias formas de comunicação. Deste modo, o Arduino é capaz de comunicar, com computadores, microcontroladores ou sensores, através de I²C, SPI ou Série.

4.1.2 Wireless SD Shield

A existência de várias *shields*, é outra das vantagens do Arduino, pois cada uma delas permite acrescentar novas funcionalidades ao modelo base de Arduino utilizado, sem que se façam alterações no circuito eletrónico do Arduino. Estes módulos são construídos de forma a encaixarem tal como “peças de Lego” na placa principal e mesmo noutras *shields*, através de um conjunto de pinos macho. Existem no mercado diversos módulos deste tipo, sendo que no caso desta dissertação, o fator de seleção recaiu no tipo de comunicação sem fios a utilizar, o Zigbee², e no armazenamento de dados num cartão de memória.



Fig. 6 - Arduino Wireless SD Shield (Fonte: Inmotion)

Deste modo, foi escolhida a Wireless SD Shield (Fig. 6), desenvolvida pela mesma empresa do Arduino, por ter entradas do tipo pino fêmea, compatíveis ao encaixe dos módulos de comunicação Xbee, além da existência de um compartimento para leitura de cartões microSD. Um dos principais detalhes desta *shield* prende-se com o facto de possuir um interruptor, capaz de seleccionar uma de duas opções possíveis:

1. USB – Na posição USB apenas é possível programar o microcontrolador, e trocar dados pela porta série.
2. Micro - A seleção da posição Micro permite comunicar o módulo Xbee ao Arduino, e trocar dados com outros módulos que suportem Zigbee. Por outro lado deixa de ser possível programar o microcontrolador através de USB.

4.1.3 Transformador de Corrente

Um transformador, seja de corrente (TC) ou de tensão (TT), define-se como sendo um equipamento capaz de alterar uma corrente ou tensão para valores ajustados a uma determinada função. No caso em concreto do TC, é capaz de medir a corrente alternada, produzindo uma corrente secundária proporcional à corrente detetada no primário. Este procedimento é bastante útil porque a corrente

² Tecnologia Zigbee implementada pela empresa Digi nos módulos Xbee. Na secção dedicada à comunicação será feita uma análise mais exaustiva a esta tecnologia.

presente nas redes elétricas é demasiado alta para ser recebida diretamente nos canais de entrada dos aparelhos de medição, neste caso o Arduino (Elkor Technologies Inc., 2006).

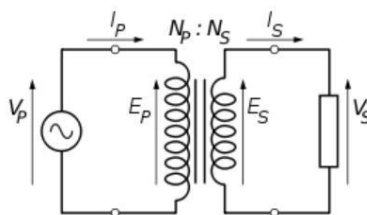


Fig. 7- Circuito Elétrico de um Transformador (Fonte: Wikipedia)

Baseando-se na Lei de Faraday e no princípio de funcionamento dos transformadores convencionais (ver Fig. 7), uma corrente alternada passa no enrolamento primário produzindo um campo magnético no núcleo, que por sua vez induz uma corrente no enrolamento secundário. Este método permite uma fácil instalação e um alto nível de isolamento entre os dois enrolamentos. Através da razão do número de espiras do enrolamento secundário e do enrolamento primário, é possível determinar a razão de transformação entre a corrente de entrada e a corrente de saída no TC.

Como já foi referido anteriormente, um transformador de corrente é constituído por dois enrolamentos e um núcleo magnético, sendo o primeiro geralmente fabricado em cobre e o segundo em aço de silício, liga de níquel ou ferrite. O material utilizado no fabrico do núcleo tem interferência na precisão do transformador, em termos da relação corrente de entrada/saída, bem como na linearidade.

A forma do núcleo é um dos aspetos mais preponderantes na decisão do tipo de transformador de corrente a utilizar, resumindo-se a dois tipos: núcleo sólido e núcleo dividido. O transformador de núcleo sólido apresenta como vantagem a não existência de um espaço vazio no núcleo permitindo uma melhor performance no que toca a precisão e deslocamento de fase. Por sua vez o transformador de núcleo dividido por ser não invasivo não exige modificações nos cabos elétricos para a sua colocação, resultando numa maior facilidade e segurança de instalação. Este pormenor é importante, quando comparado com outros tipos de transformadores de corrente existentes no mercado, como os de efeito de Hall e Bobines Rogowski que são do género invasivo.

Existe uma grande variedade de transformadores de corrente disponíveis para compra na internet, tendo neste caso a escolha recaído no CR3110-3000 (Fig. 8), da marca CR Magnetics.



Fig. 8 - CR Magnetics CR3110-3000 (Fonte: DigiKey)

Este TC apresenta como características os seguintes pontos (CR Magnetics):

- corrente máxima a medir: 75A;
- número de espiras do enrolamento secundário: 3000;
- núcleo em Ferrite.

Ao contrário de outros modelos de TC, o modelo escolhido não apresenta uma resistência de carga no seu interior, pelo que teve de ser dimensionada uma que satisfizesse os critérios requeridos. No capítulo dedicado à montagem do sistema será explicado em pormenor este assunto.

4.1.4 Sensor de Infravermelhos Passivo

O sensor de infravermelhos passivo, IVP (*Passive Infrared Sensor-PIR*) é um sensor capaz de detetar o movimento de pessoas ou animais a partir da diferença entre a radiação térmica emitida pelo corpo e o ambiente em redor. O resultado é apenas qualitativo, não sendo possível o cálculo da distância a que está o corpo. Este tipo de tecnologia parte do princípio de funcionamento dos detetores de movimento optoeletrónicos, que dependem da radiação eletromagnética no campo ótico, neste caso a que possui maior comprimento de onda (c.o.), do tipo visível e infravermelha. A radiação detetada pode ser refletida ou emitida por um corpo e é produzida por uma de duas formas, fonte externa (forma passiva) ou mesmo pelo próprio corpo (forma ativa). No caso analisado tem-se um detetor do tipo passivo, que capta a diferença de temperaturas de diferentes corpos a partir da radiação infravermelha emitida pelos mesmos (Springer, 2010).

Em detalhe, um sensor IVP é capaz de detetar trocas ao nível do calor radiativo entre o elemento sensor e o objeto em movimento. Para que isto aconteça é necessário partir da teoria da emissão de radiação eletromagnética aplicada aos objetos em que a temperatura está acima do zero absoluto. Além desta teoria, importa salientar a lei de Stefan-Boltzman que permite calcular a intensidade da radiação emitida por todos os objetos. Tendo em conta que todos os objetos emitem radiação, é essencial para o funcionamento do sensor que o objeto a detetar tenha uma temperatura de superfície diferente do ambiente onde está inserido, pois só desta forma existirá um contraste térmico.

A aplicação dos sensores IVP na deteção de pessoas é bastante comum, já que, tanto a pele humana como as fibras utilizadas na confecção de roupa têm altos valores de emissividade, 90% e 75-95% respetivamente.

Um sensor IVP é composto por um dispositivo de focagem (ex: lente de Fresnel), um elemento sensor (ex: cristal piroelétrico) e um comparador de limiar ou *threshold*. O primeiro elemento referido, o dispositivo de focagem, faz o registo da imagem presente no espaço que o sensor está a analisar, enquanto o elemento sensor converte a radiação captada num sinal elétrico. Por último o comparador tem a função de converter o sinal analógico, produzido pelo elemento sensor, num sinal digital.

No caso dos elementos sensor, existem também três tipos passíveis de serem utilizados nos sensores IVP: os termístores, as termopilhas e os piroelétricos. Apesar disso, os piroelétricos são os elementos sensor aplicados maioritariamente na deteção do movimento, devido ao seu baixo custo, alta resposta e simplicidade. Relativamente ao mecanismo de funcionamento do elemento piroelétrico, este rege-se pela produção de uma carga elétrica em resposta a um fluxo de energia térmica que passe no mesmo elemento. Assim, o calor absorvido pelo elemento sensor faz com que o seu lado frontal se expanda, resultando num stress térmico induzido. Este fenómeno aliado ao facto de todos os sensores piroelétricos serem também piezoelétricos, origina uma carga piezoelétrica nos elétrodos do elemento sensor. Contudo este piezoelétrico pode ter efeitos negativos derivado às suas características, pois também produz cargas quando é submetido a alterações mecânicas provocadas por forças externas, como por exemplo o som ou as vibrações. De forma a evitar este problema, é necessário separarem-se as cargas produzidas por cada um dos fenómenos referidos, através de um fabrico simétrico do sensor piroelétrico. A ideia base é de que, caso as interferências anteriormente mencionadas sejam aplicadas em simultâneo aos dois elétrodos, seja cancelado o sinal de entrada no circuito eletrónico. Por outro lado, a radiação térmica será absorvida por cada um dos elementos sequencialmente, impedindo que exista o tal cancelamento de sinal, denominando-se esta combinação por sensor diferencial.

No que toca ao dispositivo de focagem, a lente de Fresnel fabricada em plástico tornou-se a opção mais popular. Apresenta como principais vantagens um baixo custo, por se utilizar o plástico no seu fabrico, permitindo este material maleabilidade a qualquer formato e, além disso, desempenha um papel de barreira e proteção do piroelétrico a eventuais poluentes/poeiras vindas do exterior. Além das vantagens referidas, a lente tem a função de limitar o intervalo de transmissão da radiação infravermelha entre c.o. de 8 e 14 μ m, bastante próximos dos valores emitidos pelo corpo humano (Glolab, 2011). A sua forma convexa e de divisão por secções permitem que a radiação recebida seja focada no cristal central do sensor piroelétrico.



Fig. 9- Hanse Electronics SE-10 (Fonte: Sparkfun)

O sensor de infravermelhos passivo selecionado, foi o SE-10 (Fig. 9), da marca Hanse por ser um sensor capaz de desempenhar as funções requeridas e ter um baixo custo e consumo associado. O SE-10 apresenta como principais características:

- alcance: ≈ 6 metros, segundo os testes efetuados;
- alimentação: 12 V (também funciona a 5V);
- sinal de saída: Digital;
- material de fabrico da lente: Polietileno;
- sensor piroelétrico: LHI 878 da PerkinElmer com as seguintes características:
 - 2 elétrodos;
 - alimentação: 2 a 15 V;
 - campo de visão: horizontal-95° e vertical-90°;
 - temperatura de funcionamento: -40 a 85 °C;
 - revestimento do filtro em silicone.

Além das características presentes na folha de produto do sensor, o SE-10 e regra geral os sensores IVP têm uma saída em coletor aberto (*open collector*), o que implica que seja utilizada uma resistência *pull-up* entre o sinal produzido pelo sensor (pino alarme) e o pino de alimentação. Este aspeto, bem como, a colocação do sensor no circuito serão descritos em pormenor no sistema desenvolvido.

4.1.5 Sensor de Ultrassons

O sensor de ultrassons é um sensor capaz de calcular a distância a que se encontra um determinado corpo do sensor. O princípio básico de funcionamento deste tipo de sensores resulta da emissão de um sinal por parte do mesmo sensor e, caso este sinal encontre um objeto no seu caminho sofrerá reflexão. O sinal refletido, também referido como eco, irá voltar no sentido do sensor intersetando-o, e este através do tempo total que o sinal demorou a decorrer, calcula a distância.

Apesar de parecer um processo simples, todo este mecanismo esconde alguns passos importantes e essenciais ao sucesso da deteção de objetos e pessoas por este tipo de tecnologia. Para o descrever é necessário partir do sinal, que é utilizado durante a emissão e receção, ou seja, uma onda. Tal como o nome do sensor indica, no decorrer da deteção é utilizada uma onda do tipo ultrassom. Este género de ondas é caracterizado por uma elevada frequência e baixo comprimento de onda. Normalmente a frequência de onda produzida por este tipo de sensores encontra-se acima dos 20 kHz, o que implica que o ouvido humano seja insensível à mesma, e só animais como o morcego possam identificá-la. Para que o objeto produza eco, o seu comprimento tem de ser superior ao comprimento de onda do sinal.

A criação da onda, no caso referido, vai resultar da conversão da energia elétrica, recebida pelo sensor a partir do microcontrolador, em energia mecânica. Esta mesma energia mecânica é produzida através do movimento de uma superfície que, por sua vez, irá gerar a compressão e expansão do meio em que está inserido (Springer, 2010). A forma mais convencional de criar esta compressão/expansão resulta

da utilização de um transdutor piezoelétrico, um dos componentes do sensor. É este transdutor, que na prática existe sob a forma de um elemento cerâmico piezoelétrico, que transforma a energia elétrica em mecânica, tal como se referiu anteriormente. Deste modo, a tensão recebida é aplicada no elemento cerâmico flexível, que produz uma onda. Além disso, devido às características reversíveis do elemento cerâmico, é possível que o fenómeno contrário ocorra, e assim se converta uma onda (energia mecânica) em tensão (energia elétrica). Este aspeto é importante para a distinção dos dois tipos de sensores de ultrassons existentes no mercado, relativamente a este detalhe, os que têm uma partilha do elemento cerâmico por parte do emissor e recetor, e os sensores que os têm separados. No caso do sensor utilizado na tese, tem-se que o emissor e o recetor estão juntos, sendo este modelo utilizado, para casos em que é possível intervalar a emissão de ondas com a receção usando pulsos.

A distância entre o sensor e o objeto detetado é calculada da seguinte fórmula:

$$L_0 = \frac{v.t.\cos\theta}{2} \quad (8)$$

onde t representa o tempo que o sinal demora, desde a sua emissão pelo sensor, até ser refletido pelo objeto e voltar a ser recebido pelo mesmo sensor. O facto de o tempo englobar a distância de ida e volta do sinal, implica que se divida por dois. Além do tempo, temos v para a velocidade média da onda e θ para o ângulo.



Fig. 10 - Devantech SRF02 (Fonte: Solarbotics)

O sensor de ultrassons escolhido foi o SRF02 da marca Devantech (Fig. 10), tendo como fatores positivos, o preço reduzido, a existência de informação relativamente à sua programação, e utilização do protocolo de comunicação I²C. Relativamente às principais características técnicas tem-se que:

- Alcance de deteção: 16cm a 6m;
- Alimentação: 5V;
- Frequência: 40KHz;
- Modos de comunicação: I²C e série;
- Não é necessária calibração;
- Unidades de medida: uS, mm ou polegadas;
- Tempo de cálculo da distância: 70 ms.

O alcance mínimo do sensor é limitado pelo tempo que demora a emitir um pulso e a eliminar dados residuais que ainda possam existir, enquanto o alcance máximo é limitado pela potência dos pulsos, já que a partir de uma determinada distância os ecos são demasiados fracos para serem detetados com precisão (Acroname Robotics, 2011).

4.1.6 Relógio de Tempo Real

Não obstante na descrição do Arduino ter sido referida a existência de um oscilador (*clock*), este apenas permite estabelecer um padrão de frequência responsável pelo controlo dos ciclos eletrónicos

da própria placa. Contudo no caso de se pretender uma sincronização dos dados que se estão a obter com o tempo real, é necessária a instalação de um relógio de tempo real (*Real Time Clock*, RTC). Este género de relógios está presente na maioria dos dispositivos eletrónicos (computadores, sistemas embebidos), possibilitando um controlo preciso do tempo, e nas unidades utilizadas pelos humanos. Tal como no Arduino, também no caso do RTC é utilizado um oscilador de cristal (normalmente quartzo) para fazer esta gestão do tempo. Apesar da elevada precisão destes osciladores são frequentes atrasos ou adiantamentos por parte dos RTC, sendo estes fenómenos causados por alterações de temperatura ou campos magnéticos.



Fig. 11- Maxim DS1307

O RTC escolhido foi o DS1307 (Fig. 11) da marca Maxim. Este modelo vem com todos os componentes necessários ao seu funcionamento montados, além de estar pré-programado pelo fabricante. De forma a poder funcionar, este relógio contém um oscilador formado por um cristal de quartzo com uma frequência de 32,8 kHz, sendo alimentado por uma pilha de lítio que permite um funcionamento contínuo de pelo menos 9 anos, sem qualquer tipo de alimentação externa. Além destes detalhes técnicos o modelo DS1307 possui as seguintes características:

- Comunicação via I²C;
- Contagem dos segundos, minutos, horas, dias da semana, dias do mês, mês, ano (incluindo anos bissextos);
- Caso falhe a alimentação de energia por outra fonte que não a pilha, passa a ser automaticamente alimentado por esta;
- Consumo inferior a 500nA, quando alimentado por pilha;
- RAM de 56 bytes com capacidade para escritas infinitas.

Em relação ao tempo de atraso/adiantamento deste modelo, é de esperar que não ultrapasse o de um minuto por mês.

4.2 Software

O software utilizado engloba os dois programas criados, no caso do Arduino IDE para programação do microcontrolador Arduino, e no caso do Processing para criação de uma interface gráfica de registo dos valores obtidos, bem como armazenamento dos mesmos e exportação para a Internet.

4.2.1 Processing

O Processing é definido como uma linguagem de programação e ambiente de desenvolvimento integrado (IDE) próprio para trabalhos na área das artes visuais e gráficas. Este IDE foi desenvolvido pelo MIT, tendo como principais características o facto de ser gratuito, uma alternativa em *open source* a vários softwares existentes no mercado bastante caros, e de permitir a aquisição dos conceitos de programação, mesmo para pessoas fora da área da informática (Processing, 2012). Em relação à linguagem utilizada, o Processing é construído sobre Java, utilizando uma sintaxe simples, que permite abstrair da complexidade da própria linguagem Java.

No contexto da dissertação a utilização do Processing serviu para desenvolver uma interface gráfica capaz de ser reproduzido no computador, e que pudesse reportar os dados monitorizados em tempo

real. Outro dos aspetos importantes para a escolha do Processing, foi a semelhança da linguagem com a do Arduino IDE.

4.2.2 Arduino IDE/Wiring

Qualquer definição do Arduino IDE não pode ser feita sem uma menção ao projeto Wiring, confundindo-se estas duas aplicações, devido à elevada semelhança. O Wiring foi desenvolvido sobre o Processing mantendo a mesma filosofia de código aberto, tendo-se distinguido como uma linguagem própria para programação de microcontroladores (Wiring, 2012). Assim, quando se fala em Arduino IDE, pode-se definir como sendo a linguagem Wiring aplicada ao caso específico do Arduino. Ambos os ambientes de desenvolvimento são escritos em Java, contudo incluem uma biblioteca C/C++, também denominada Wiring, que simplifica as operações de entrada e saída no Arduino, abstraindo o utilizador do baixo nível da linguagem C/C++ (Wikipedia, 2012).

Tendo uma interface de desenvolvimento idêntica à do Processing, no Arduino IDE escreve-se o código, e através do campo “Serial Monitor”, indica-se a porta onde está ligado o Arduino a carregar com o código desenvolvido.

4.3 Comunicação

No tópico da comunicação são abordadas as quatro formas de comunicação dos dados. No caso do USB é utilizado para fazer comunicação e alimentação ao Arduino e aos módulos Xbee. O I²C é utilizado para trocar dados entre o ultrassons e o RTC com o Arduino, através de ligação com fios. O SPI é utilizado para se realizar a comunicação com o cartão micro SD. Por sua vez o ZigBee é uma tecnologia que permite trocar dados, através de comunicação sem fios, entre sensores, microcontroladores e/ou computadores.

4.3.1 USB

Uma das formas mais básicas de comunicação entre dispositivos periféricos e computadores é a partir de ligação USB (*Universal Serial Bus*). No caso da dissertação o USB foi utilizado, na programação do Arduino, sendo a única forma de carregar no microcontrolador o *sketch* desenvolvido no Arduino IDE. Além disso foi necessária para configuração de ambos os módulos Xbee, e ligação do módulo Xbee que desempenha as funções de coordenador, na receção de dados lidos pela interface (ver secção Configuração da Rede ZigBee). O USB foi criado por grandes empresas na área eletrónica de forma a padronizá-lo como forma de comunicação de periféricos, a baixo custo, fácil utilização e capaz de atingir altas velocidades (Wikipedia, 2012). De forma simples, é definido como sendo um barramento série externo, possuindo quatro contactos, dois deles para transmissão dos dados e os restantes dois para a alimentação de energia.

4.3.2 I²C

Como se verificou na descrição dos dispositivos utilizados, no caso do ultrassons e do relógio de tempo real, ambos utilizam o protocolo de comunicação I²C (*Inter-Integrated Circuit*). O I²C é um barramento série utilizado para ligar periféricos de baixa velocidade a uma placa mãe, tendo sido criado pela Philips. Tal como é apresentado na montagem do sistema (Fig. 17), a comunicação entre periféricos e placa mãe é feita a partir de duas linhas:

- SDA (*Serial Data Line*), responsável pela transmissão dos dados;
- SCL (*Serial Clock Line*), responsável pela transmissão de impulsos a um intervalo regular, ou seja, o sinal de relógio (*clock*).

A principal vantagem associada à utilização do I²C, prende-se com o facto de permitir que um máximo de 112 escravos (ex: sensores) possam ser controlados por um único mestre (*Master*), utilizando apenas as duas linhas anteriormente referidas. A solução encontrada para distinguir por escravo (*Slave*) os dados trocados, passa pela utilização de diferentes endereços para cada um dos dispositivos. No caso do Arduino apenas são utilizados os dois pinos analógicos 4 e 5, para a utilização do I²C.

Além da vantagem anterior, o I²C comunica de uma forma digital, permitindo que dados que seriam enviados analogicamente possam desta forma ser enviados digitalmente.

Em relação à forma como são trocados os dados, resumidamente, sempre que o relógio regista alterações de nível baixo para alto, os dados pretendidos são obtidos a partir de um dado endereço do dispositivo requerido. Desta forma é transferida a informação do Arduino para o dispositivo I²C usando a linha SDA, com o endereço de onde se pretendem obter os dados do dispositivo. Quando o pino relógio muda de alto para baixo, o dispositivo chamado transmite os dados pretendidos de volta para o Arduino através da mesma linha.

Como o código por de trás da implementação do protocolo de comunicação I²C é de baixo nível, aproveitaram-se os recursos disponibilizados para o seu funcionamento no Arduino, que consistiram na utilização da biblioteca (“Wire”) própria para I²C, que permite a abstração de toda a complexidade do código de comunicação.

4.3.3 SPI

O protocolo *Serial Peripheral Interface* (SPI) é utilizado no sistema desenvolvido para comunicação entre o cartão micro SD e a *shield*, ou seja, é indispensável para guardar os dados produzidos pelo Arduino. O SPI é definido como sendo um protocolo de sincronização de séries de dados, usado por microcontroladores para comunicar rapidamente com um ou mais dispositivos periféricos a curtas distâncias (Arduino, 2012).

Tal como o protocolo I²C também o SPI utiliza o sistema mestre e escravo. Assim, é indispensável a definição de um dos dispositivos como mestre, no caso o microcontrolador, que controla os restantes dispositivos. A comunicação entre o microcontrolador e os dispositivos é feita a partir de três linhas:

- MISO (*Master in Slave Out*), linha do escravo para envio de dados ao mestre
- MOSI (*Master Out Slave In*), linha do mestre para envio de dados aos dispositivos
- SCK, responsável pela transmissão dos impulsos do relógio para sincronização dos dados produzidos pelo mestre

Para estabelecer a ligação entre o microcontrolador e o dispositivo por SPI é necessário seleccionar o pino do escravo (*Slave Select-SS*), que o mestre utiliza para ativar o dispositivo. No Arduino Uno são utilizados o pino 10 para SS, o pino 11 para MOSI, o pino 12 para MISO e o pino 13 para SCK. Tal como no I²C, também para o SPI, no caso concreto da ligação ao cartão micro SD, existe uma biblioteca (“SD”) para o Arduino.

4.3.4 A Tecnologia ZigBee

Um dos objetivos requeridos na realização desta dissertação foi a utilização da comunicação sem fios para transferência de dados entre o Arduino e o computador, de forma a evitar problemas com eventuais distâncias entre os mesmos dispositivos. Não obstante a existência de várias tecnologias de comunicação plausíveis de serem utilizadas, tais como Wi-Fi e Bluetooth, a escolha recaiu na ZigBee.

O protocolo ZigBee surgiu a partir de uma das várias normas definidas pelo grupo de trabalho IEEE 802.15 (IEEE-*Institute of Electrical and Electronics Engineers*). Este grupo é responsável pela definição e controlo das normas a utilizar nas chamadas WPANs (*Wireless Personal Area Networks*), ou seja, nas redes de área pessoal sem fios de curto alcance na ordem dos 10 metros. No caso específico do Zigbee a norma definida é a 802.15.4, enquanto por exemplo o Bluetooth é definido pela norma 802.15.1. O Zigbee foi desenvolvido de forma a cumprir uma série de requisitos, tais como, baixo custo de implementação da rede, reduzido consumo energético, e baixa velocidade de transferência de dados (ZigBee Alliance, 2012). Este conjunto de características tem contribuído para um aumento da utilização do ZigBee em redes de sensores e transmissão de dados, tal como acontece neste caso, sendo a automação de casas e monitorização de consumos, uma das principais áreas de aplicação.

Para entender a forma como o ZigBee se interliga com a norma IEEE 802.15.4 é necessário compreender a arquitetura do protocolo e das várias camadas que o constituem. Partindo do modelo OSI (*Open System Interconnection*), são definidas sete camadas de abstração distintas, sendo estabelecida uma hierarquia do nível inferior para o nível superior. Deste modo, as funcionalidades de uma determinada camada são servidas pelas das camadas inferiores e, por sua vez, passam para as camadas superiores (Wikipedia, 2012). No caso do ZigBee, das sete camadas que compõem o modelo OSI, apenas são utilizadas as quatro camadas relevantes, que lhe permitem alcançar as funcionalidades pretendidas. A Fig. 12 representa a arquitetura da pilha de protocolos do ZigBee.

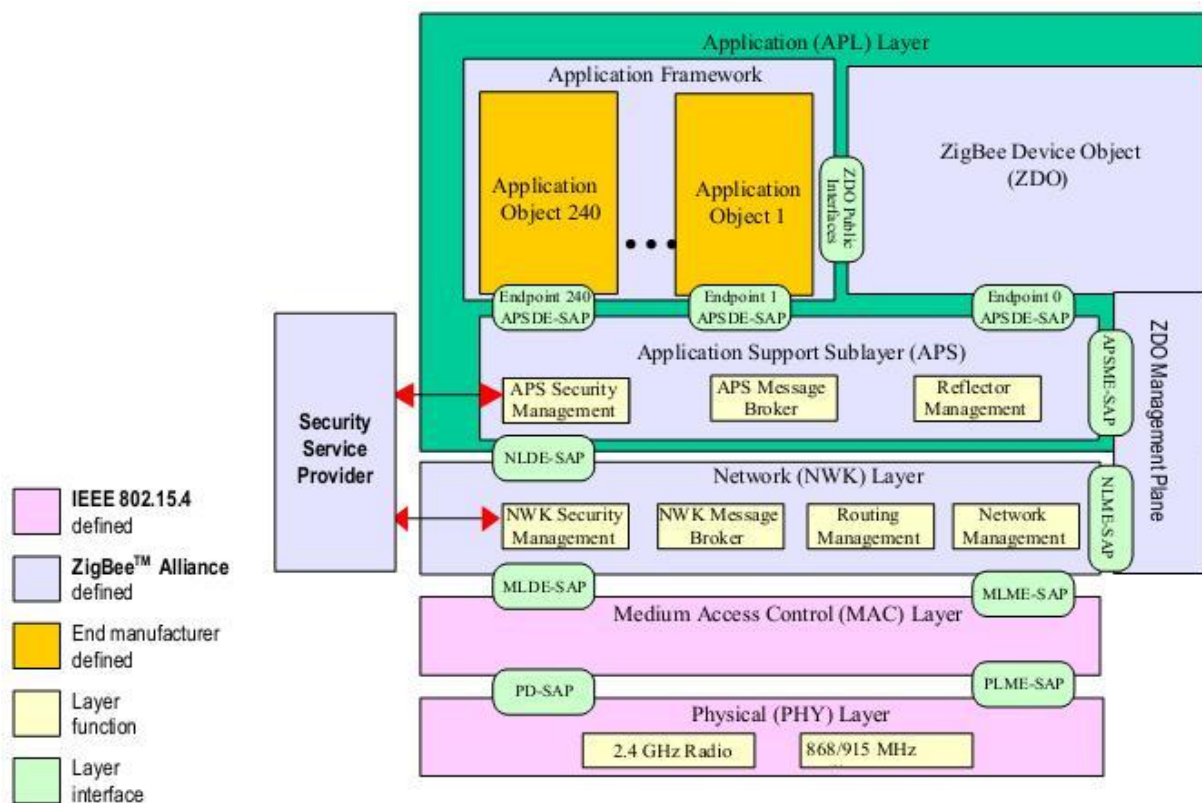


Fig. 12 - Arquitetura do Protocolo ZigBee (ZigBee Standards Organization, 2006)

As duas camadas inferiores da pilha, camada física ou PHY (*Physical layer*) e camada de controlo de acesso ao meio ou MAC (*Medium Access Control*), são definidas pela norma IEEE 802.15.4. Por outro lado as restantes duas camadas, camada de rede ou NWK (*Network*) e camada de aplicação ou APL (*Application layer*), são definidas pela ZigBee Alliance.

4.3.4.1 IEEE 802.15.4

A camada física do protocolo, gerida pela norma IEEE 802.15.4, permite a interligação com o rádio de comunicação. Na Fig. 12 observa-se que a camada física pode operar através de dois intervalos de frequência não licenciados distintos, um de 2,4 GHz e outro de 868/915 MHz. Estas frequências de banda apresentam três valores distintos, sendo que, cada um deles destina-se a uma determinada região do planeta e tem valores de velocidade de transmissão diferentes. A frequência de 868 MHz pode ser apenas utilizada no espaço Europeu e conta com uma velocidade de transmissão de 20 kbps, enquanto a de 915 MHz pode ser usada em países como os EUA e a Austrália e tem uma velocidade máxima de 40 kbps. Por último a frequência de 2,4 GHz é utilizada à escala mundial e permite uma velocidade de transmissão de 250 kbps.

A camada MAC tem como principais funções o controlo do acesso ao canal rádio e processamento de segurança da norma. A norma IEEE 802.15.4 estipula dois tipos de dispositivos distintos: o FFD (*Full Function Device*) e o RFD (*Reduced Function Device*). O primeiro é um dispositivo funcional capaz

de comunicar com qualquer um dos dispositivos que compõem a rede, tendo a função de criar e coordenar a rede, além de conseguir desempenhar qualquer papel em termos de elemento de rede, atuando na maioria das vezes, como coordenador ou como router. O FFD desempenha um papel importante na segurança da rede através do armazenamento das chaves de segurança. Por outro lado, o RFD é um dispositivo de funções reduzidas, sendo bastante mais limitado que o FFD. O RFD apenas desempenha as funções de dispositivo final, podendo ligar-se a apenas um FFD de cada vez. O RFD é utilizado nas redes sem fios para conexão de sensores ou atuadores, permitindo como única tarefa, a transmissão de dados para um coordenador ou router através de um baixo consumo energético.

Topologia de rede

Na tipologia de rede existem três tipos de funções/nodos: coordenador, *router* e dispositivo final. Apesar destes conceitos serem similares aos de FFD e RFD, traduz-se como coordenador a raiz da rede, permitindo a ligação através dele a outras redes, os *routers* como dispositivos capazes de transmitir dados de uns nodos para outros e os dispositivos finais como sendo apenas capazes de enviarem informação para um coordenador ou router.

A forma como estes três elementos se associam resulta em diferentes tipologias de rede, existindo no caso desta norma três mais relevantes: estrela, árvore e ponto-a-ponto.

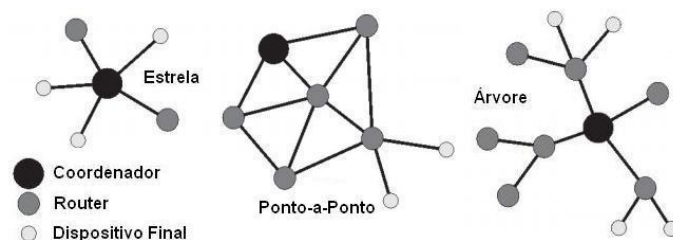


Fig. 13 - Topologias de rede para a norma IEEE. 802.15.4 (SAFARIC, et al., 2006)

A partir da Fig. 13 é possível observar as principais características de cada uma destas topologias de rede. No caso da topologia estrela a comunicação é feita entre os dispositivos (*routers*, dispositivos finais) e um único coordenador central. Na topologia ponto-a-ponto, também existe um elemento coordenador, contudo este já não tem uma função central, pois um dispositivo pode ter a liberdade de comunicar com qualquer outro não importando a sua categoria. A topologia em árvore é construída a partir da de ponto-a-ponto, sendo a maioria dos elementos formados por *routers* ou coordenadores. Nesta topologia em específico qualquer destes dois elementos pode operar como elemento coordenador, controlando a rede (IEEE, 2006).

Apesar de toda esta capacidade que a norma IEEE. 802.15.4 permite em termos de topologia, no caso estudado apenas foi utilizada uma forma bastante simples de ponto-a-ponto utilizando um coordenador e um router.

4.3.4.2 ZigBee

Como já foi anteriormente mencionado, o ZigBee acrescenta às camadas definidas pela norma IEEE duas novas camadas de nível mais alto, a de rede e a de aplicação (ver Fig. 12).

A principal função da camada de rede é de permitir uma utilização correta da camada MAC, disponibilizando à camada de nível superior uma interface apropriada. Além desta função, é responsável pela gestão das entradas e saídas de dispositivos na rede, segurança dos pacotes trocados, pelo cálculo e manutenção dos caminhos mais curtos entre dispositivos, e armazenamento de informação importante dos nós vizinhos. Além destas propriedades, a camada de rede de um coordenador ZigBee é responsável por iniciar uma nova rede, sempre que seja apropriado, gerando endereços aos novos dispositivos associados.

No que concerne à camada de aplicação do ZigBee, pela Fig. 12 é possível verificar que esta é constituída por: subcamada de suporte de aplicação ou APS (*Application Support Sub-Layer*), enquadramento de aplicações ou AF (*Application FrameWork*), ZDO (*ZigBee Device Object*) e

objetos de aplicação definidos pelo fabricante. A subcamada de suporte tem como tarefas, a manutenção de tabelas para *binding* (capacidade de emparelhar dois dispositivos a partir dos seus serviços e necessidades) e reencaminhamento de mensagens entre dispositivos vizinhos. A ZDO possibilita a definição do papel de cada um dos dispositivos na rede, como coordenadores, *routers* ou dispositivos de fim. Além desta definição dá início/resposta aos pedidos de *binding*, estabelece associações seguras entre os dispositivos e também procura novos dispositivos na rede determinando qual o serviço que providenciam.

4.3.4.3 Módulos Xbee

Devido às vantagens do protocolo ZigBee para redes sem fios de sensores, uma série de fabricantes tem vindo a desenvolver dispositivos capazes de satisfazer as necessidades do mercado. No caso desta dissertação por ser utilizado o Arduino, foram escolhidos os módulos Xbee, desenvolvidos pela empresa Digi International, capazes de se acoplarem à *shield* e Explorer USB³ adquiridos. Como se pode visualizar pela Fig. 14, a principal diferença entre os dois módulos encontra-se no tipo de antena utilizada.



Fig. 14 – Digi International Xbee (Fonte: Sparkfun)

O módulo da esquerda corresponde ao modelo XB24-Z7WIT-004, apresentando uma antena de fio, enquanto o módulo à direita corresponde ao modelo XB24-Z7CIT-004, tem uma antena de chip. No entanto, apesar das diferenças ao nível das antenas ambos os modelos correspondem à *Series 2* de módulos da Digi, tendo as seguintes características técnicas:

- Alcance: até 40 m em espaço fechado e até 120 m em ambiente exterior
- Taxa de Transmissão: 250 kbps
- Alimentação: 2,1 - 3,6V
- Corrente: 40 mA de corrente de transmissão e 40 mA de corrente de receção
- Frequência: 2,4 GHz
- Topologias de Rede suportadas: Ponto-a ponto, Ponto-a-multiponto.
- N° de canais: 16 sequências diretas de canais, e canais de 11 a 26

4.3.4.4 Configuração da Rede ZigBee

Uma das etapas essenciais, para que se possam ligar os Xbees adquiridos ao Arduino e ao PC, passa pela programação dos próprios módulos. Cada módulo Xbee possui um microcontrolador interno, que executa um programa, conhecido como *firmware*. Este programa executa por sua vez todo endereçamento, comunicação, segurança e funções úteis para o módulo (Faludi, 2011). A empresa Digi disponibiliza a aplicação X-CTU, capaz de alterar ou atualizar o *firmware*. Através da utilização de um dos módulos Xbee em conjunto com o Explorer USB liga-se este ao computador e inicia-se o X-CTU. A partir daqui é possível localizar a porta à qual está ligado o dispositivo e quais as suas

³ Adaptador próprio para encaixe de módulos Xbee, que se liga por mini USB a um PC, permitindo acesso de dados pela porta série e programação dos pinos do módulo.

caraterísticas e funções atuais. Para que os módulos possam comunicar entre si é necessário definir primeiro qual deles desempenhará a função de coordenador da rede. No caso da dissertação, este papel coube ao Xbee com antena de fio. Consequentemente o módulo com antena de chip desempenha as funções de *router*.

Além da definição da função de cada módulo Xbee na rede sem fios, é necessário definir a forma como se rege a interação entre os rádios e o microcontrolador/computador. Existem dois modos diferentes de realizar esta interação, através de modo transparente (AT) ou de modo comando (API- *Application Programming Interface*). No caso do modo transparente é utilizado como forma direta de interação entre Xbee e microcontroladores/computadores, sendo o modo pré-definido de todos os Xbee. Esta configuração é bastante simples, permitindo que os rádios transmitam de uma forma direta, para a porta série, a informação que lhes chega. Desta forma, não existe qualquer alteração da formatação original dos dados trocados, contudo, devido ao seu baixo nível de robustez, é uma forma bastante ineficiente de comunicar com computadores.

Por outro lado, o modo comando é bastante mais complexo à perceção humana, havendo neste caso alteração da forma como os dados estão organizados. Esta alteração é feita para que os dados trocados sejam bastantes mais estruturados e perceptíveis aos dispositivos eletrónicos que interagem com os rádios. Assim cria-se um protocolo de interação, na camada de aplicação do Xbee, formando uma sequência de dados (*frames*), que incluem bytes capazes de identificar a sequência de dados em termos do seu início, tamanho, tipo e soma total.

No sistema desenvolvido utilizou-se o primeiro modo de interação devido à simples configuração dos módulos XBee que esta requeria. Sendo assim, após a definição das duas características anteriormente mencionadas (função e modo de interação dos módulos) foi possível utilizar o XCTU para configurar uma série de outros parâmetros comuns aos dois módulos para que se pudesse estabelecer a RSSF (Rede de Sensores sem Fios).

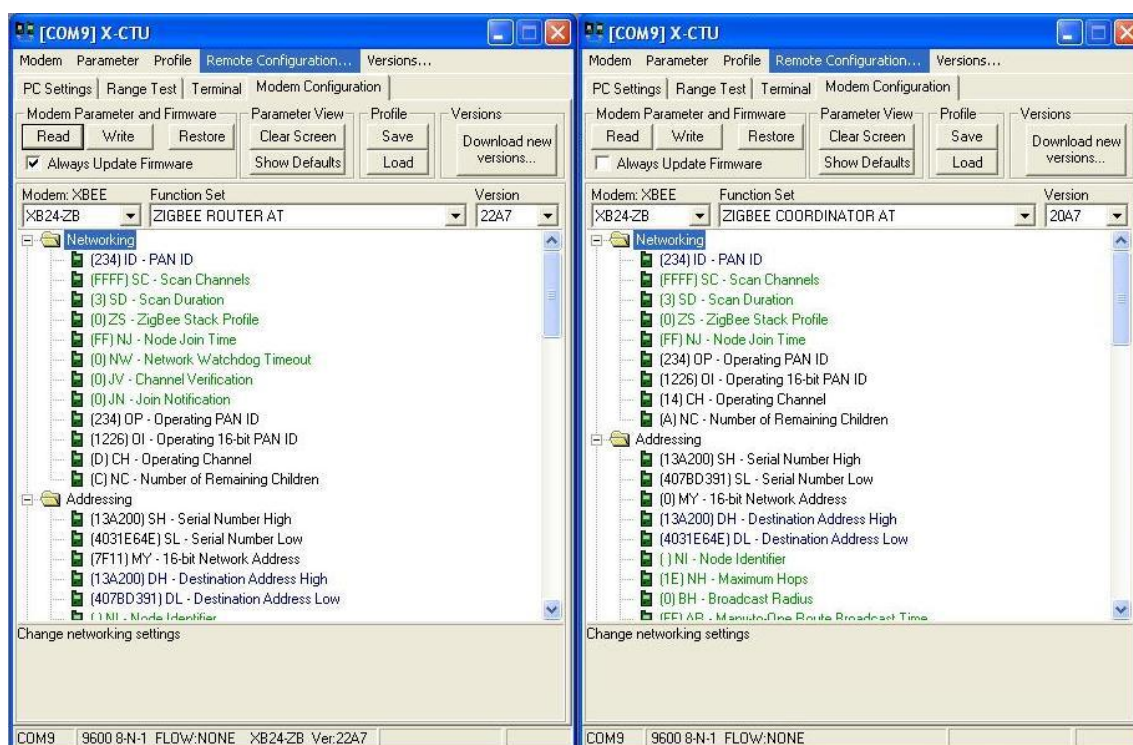


Fig. 15 - Configurações dos módulos utilizados, à esquerda a do Router e à direita a do Coordenador

Como se pode ver pelas imagens da configuração dos módulos utilizados (Fig. 15), apesar de existirem bastantes possibilidades em torno dos parâmetros a configurar, apenas alguns foram relevantes na tese, sendo eles os seguintes:

- PAN ID: Endereço PAN (*Personal Area Network*) de 16-bit. Existem 65536 endereços PAN diferentes passíveis de serem utilizados.
- Channel: ao ser detetado um novo endereço PAN, é verificado o número de canais disponíveis, normalmente 12. Para que se possa criar uma RSSF é necessário que todos os rádios usem o mesmo canal. No caso da versão do Firmware utilizada a escolha do canal é automática.
- Destination Adress High/Low (DH or DL): O DH corresponde ao número de série superior do módulo de destino dos dados enviados, enquanto o DL indica o número de série do módulo de destino inferior. Na configuração do *router* tanto o DH como o DL correspondem aos respetivos números de série do módulo coordenador, enquanto na configuração do coordenador se observa o caso oposto.

4.4 Internet das Coisas (IoT-Internet of Things)

Um dos principais aliciantes do desenvolvimento do sistema de monitorização foi a possibilidade de integrar no protótipo o conceito bastante recente da Internet das Coisas ou *Internet of Things*. Podendo ser entendida como uma nova revolução tecnológica, a Internet das Coisas é apresentada como sendo a chave para o futuro da computação e comunicação, estando diretamente ligada à área das redes de sensores sem fio e da nanotecnologia (Kranengurg, 2008).

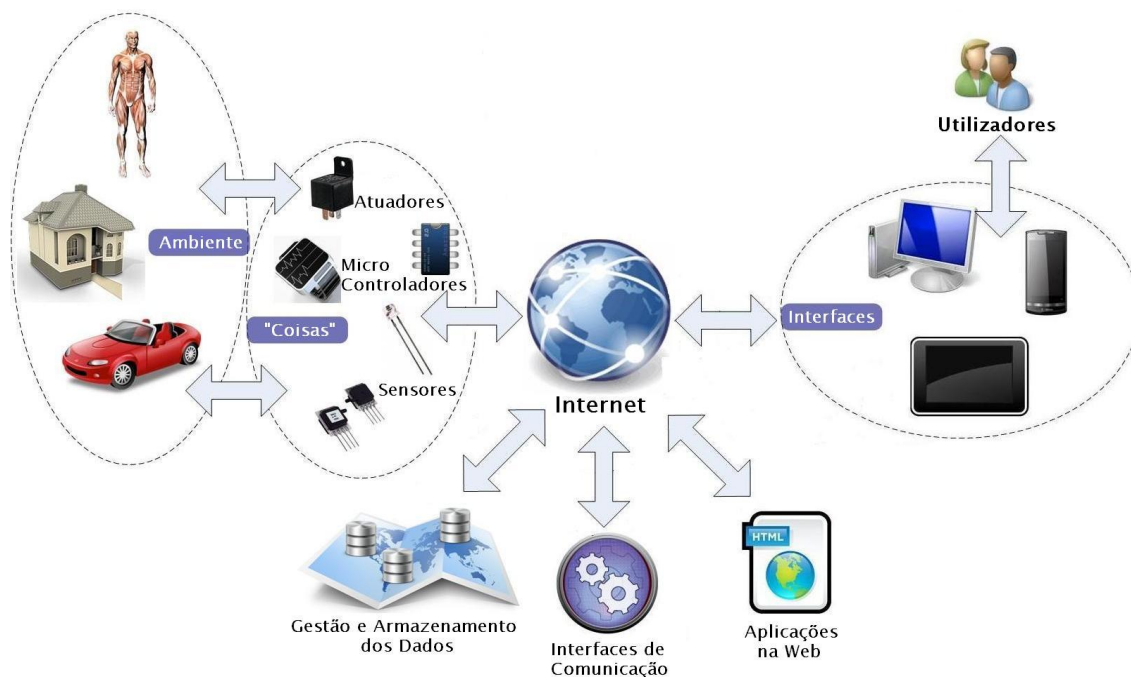


Fig. 16 - Representação da estrutura da Internet das Coisas (Fonte: Doukas, 2012)

Como se pode ver pela Fig. 16, o conceito de Internet das Coisas parte da interligação de quatro setores principais com a Internet:

- ambiente a analisar;
- suporte físico que permite medir e/ou atuar no ambiente;
- interface que permite mostrar o que está a acontecer no ambiente;
- comunicação e gestão dos dados.

Nas várias características do sistema proposto já mencionadas foram tratadas estas quatro áreas da Internet das Coisas, faltando a partilha dos dados em tempo real na Internet. O facto de existirem algumas plataformas já desenvolvidas e disponibilizadas gratuitamente para este tipo de utilização, fez

Partindo do esquema da Fig. 17, importa fazer uma análise em particular aos circuitos utilizados para cada um dos sensores.

Transformador de corrente

Para receber o sinal medido pelo sensor de corrente no Arduino foi necessário introduzir uma série de componentes eletrónicos para que a corrente medida na linha de fase ou de neutro não danificasse as portas de entrada do ADC. Como foi dito na descrição do hardware, o Arduino está limitado a um intervalo de tensão a receber nas portas analógicas, sendo que apenas pode receber tensão positiva até um máximo de 5 volts.

O primeiro componente introduzido no circuito do TC foi uma resistência de carga (R_1), sendo este um elemento essencial para o bom funcionamento tanto do transformador como do Arduino, pois como foi dito as portas de E/S analógicas recebem os valores sobre a forma de um sinal em tensão. A resistência de carga permite fechar o circuito do sistema secundário e desta forma produzir um sinal de tensão útil que seja proporcional à corrente do enrolamento secundário. Para que isto aconteça é necessário que a resistência seja corretamente dimensionada, o que implica, por outro lado, que não haja saturação do núcleo magnético do sensor. O valor da resistência foi dimensionado a partir do valor máximo de corrente eficaz que o TC pode medir no primário, 75A (Eq. 9).

$$I_{\text{primária_de_pico}} = I_{\text{rms}} \times \sqrt{2} = 106 \text{ A} \quad (9)$$

Sabendo que o TC utilizado tem um número de espiras igual a 3000, é possível calcular a corrente no enrolamento secundário:

$$I_{\text{secundária_de_pico}} = \frac{I_{\text{primária_de_pico}}}{N} = 0,034 \text{ A} \quad (10)$$

Com o objetivo de maximizar a resolução da monitorização e, tendo em conta, que a tensão à saída da resistência de carga será uma curva com amplitude de 2,5V, calcula-se a resistência de carga ideal. Este cálculo utiliza a corrente secundária de pico e metade do valor de tensão de referência do Arduino, 5V (Eq. 11).

$$R_{\text{carga}} = \frac{\frac{V_{\text{referência}}}{2}}{I_{\text{secundária_de_pico}}} = 71 \Omega \quad (11)$$

No caso do sistema montado, devido à inexistência de resistências com o valor calculado, optou-se pela utilização de uma resistência de 75 ohms. Em relação ao limite máximo de tensão o dimensionamento anterior e o número de espiras do enrolamento secundário asseguram que a tensão será sempre inferior aos 5 volts. Por outro lado, o facto da corrente medida ser alterna resulta num sinal com uma componente negativa, que não pode ser lida pelo Arduino. Assim, no circuito em causa a forma de condicionar o sinal para que apenas cheguem valores de tensão positivos aos pinos analógicos passou pela introdução de um divisor de tensão. Apesar de existirem outros tipos de circuitos de retificação (ex.: circuito de retificação de meia onda/onda completa), o divisor de tensão tem a vantagem de ser uma técnica bastante mais simples.

A regra do divisor de tensão consiste em colocar um conjunto de resistências em série (R_2 e R_3), sendo produzida uma tensão elétrica de saída proporcional à tensão de entrada. Através das resistências colocadas no circuito e utilizando a fórmula do divisor de tensão:

$$V_{\text{saída}} = \frac{R_2}{R_2 + R_3} \times V_{\text{entrada}} \quad (12)$$

é possível determinar a tensão de saída. Caso se ligue o TC (já com a resistência de carga em paralelo) à massa e à alimentação de 5V, pelas variações de corrente no primário do TC, observa-se que os valores oscilam em torno de zero no intervalo de $[-2,5; +2,5]$ V. Porém após a introdução de duas resistências iguais, ligadas à alimentação e à massa (ver Fig. 17), forma-se o divisor de tensão, tendo à sua saída 2,5V. Este valor resulta numa translação do intervalo original para valores entre $[0; +5]$ V, passando a oscilar em torno de 2,5V. O valor de R2 e R3 utilizado foi de 10 k Ω cada.

Além da resistência de carga e do divisor de tensão foram também colocados dois díodos zener (D1 e D2) de 22 volts cada e um condensador (C1) de 10 μ F. A colocação dos díodos tem o intuito de dar maior proteção ao material, limitando a tensão de saída caso se desconecte acidentalmente o transformador da resistência de carga. Por sua vez a colocação do condensador tem como objetivo a absorção de eventuais ruídos produzidos pelo TC, tornando os resultados finais mais fidedignos.

Pelo esquema global do sistema desenvolvido é possível visualizar que o sensor de corrente comunica o sinal recebido através da porta analógica A0 do Arduino, recebe 2,5 volts de tensão (5 volts antes da passagem no divisor de tensão) e está ligado à massa.

Sensor de Infravermelhos Passivo

Cada um dos dois sensores SE-10 utilizados apresentam três fios de funções distintas, um para alimentar o sensor através dos 5V fornecidos pelo Arduino, um para ligar à massa e outro que conduz o sinal produzido pelo sensor. No caso da transmissão do sinal produzido pelos IVP, este é recebido nos pinos digitais D2 e D3 do Arduino.

Como é possível observar pelo esquema eletrónico global do projeto, não basta ligar diretamente o sensor à tensão de saída e ao pino digital. É necessária a colocação de uma resistência *pullup* (R4 e R5), neste caso de 10k Ω , entre o fio que alimenta o sensor e o fio que transporta o sinal produzido. A introdução desta resistência está relacionada com o facto do circuito ter uma saída em coletor aberto, ou seja, sempre que não houver produção de um sinal pelo sensor o mesmo não estará ligado a nada, assegurando uma tensão de referência igual a 5V. Por outro lado, quando for detetado algum movimento pelo sensor, ele liga-se à terra. A utilização da resistência permite que, quando não haja deteção de movimento, se produza um sinal HIGH, evitando da mesma forma que haja uma flutuação aleatória do valor lido entre HIGH e LOW (Bildr, 2011).

Sensor de Ultrassons

O sensor SRF02 apresenta cinco canais para efetuar as respetivas ligações elétricas. Cada um destes canais, segundo a folha do produto, desempenha uma determinada função: Vcc a alimentação do sensor, SDA a comunicação dos dados, SCL a comunicação dos pulsos do relógio, Mode para comunicação série e GND a ligação à massa. No caso deste projeto devido à utilização da comunicação I²C não foi utilizado o canal Mode para qualquer ligação.

Em relação às ligações entre o sensor e o Arduino, o cabo SDA foi conectado à porta analógica A4, enquanto o cabo SCL foi conectado à porta analógica A5. Em relação aos restantes dois cabos o da massa foi ligado à respetiva massa do Arduino, e o de alimentação foi conectado a um condensador (C2) de 100 μ F, colocado na placa de ensaio em paralelo com o sensor. A presença deste condensador tem o objetivo de evitar eventuais flutuações do fornecimento de energia ao sensor (Arduino, 2006).

Relógio de Tempo Real

O RTC DS1307 é composto por cinco canais, sendo as ligações idênticas às do ultrassons, uma vez que ambos comunicam por I²C. Neste caso o canal SQW não é utilizado, sendo o canal SDA e SCL ligados em série à conexão estabelecida entre os mesmos canais do ultrassons às portas analógicas do Arduino A4 e A5 (ver Fig. 17).

4.5.2 Código Arduino IDE

O código desenvolvido em Arduino IDE, teve como objetivo a comunicação com os vários sensores, utilizando o microcontrolador como intermediário dos dados a receber e a enviar pela porta série, além de guardar no cartão SD. O diagrama de fluxo de toda a rotina do código desenvolvido pode ser sintetizado pelo seguinte fluxograma (ver Anexo B.):

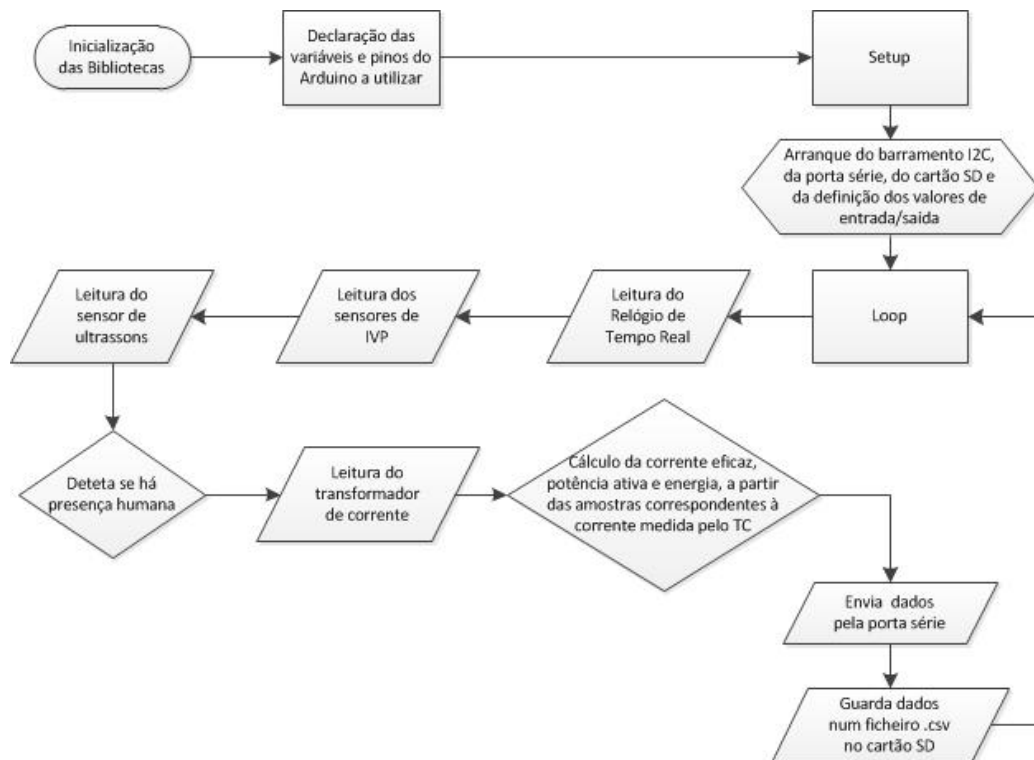


Fig. 18 - Fluxograma do *sketch* utilizado na programação do Arduino

Baseando-se na estrutura normal do código desenvolvido pelo Arduino IDE, define-se um cabeçalho inicial, onde estão indicadas as bibliotecas, variáveis e pinos a utilizar e duas rotinas indispensáveis, o Setup() e o Loop(). Dentro do Loop() encontram-se funções específicas para receber e tratar os dados de cada sensor.

No caso do sistema desenvolvido importa explicar o princípio de funcionamento do algoritmo de transformação dos dados provenientes de cada sensor.

Quanto aos sensores utilizados para a monitorização da presença foi necessário intersear os dados provenientes de cada um e enviar um único valor indicando se havia ou não presença. No caso dos IVP o valor 1 (HIGH) indica que o sensor não deteta movimento de corpos, enquanto que 0 identifica o caso oposto. Por outro lado, o sensor de ultrassons apenas mede a distância a que está um corpo, pelo que a solução encontrada para que detete movimento de pessoas, passa pela introdução do cálculo da variação da distância. Assim, a deteção de presença acontece quando os dois sensores IVP detetam em simultâneo o movimento de um corpo, ou então, caso o sensor de ultrassons detete uma variação da distância de um corpo superior a 15cm. Sempre que há deteção de presença é produzido o número 1 e, caso contrário, é produzido um 0.

Em relação à monitorização do consumo de energia elétrica o principal desafio passa por transformar simples amostras dos níveis de tensão do Arduino, para valores de corrente eficaz que, por sua vez, permitem o cálculo da potência e da energia consumida. Para que tudo isto aconteça é necessário que o sistema consiga ler rapidamente as variações de corrente no enrolamento primário do TC. Assim, é necessário definir um ciclo de amostragem capaz de calcular a corrente eficaz através da equação (3), sendo o número de amostras de cada ciclo dependente da frequência da rede elétrica. Como em Portugal a frequência é de 50 Hz, o período de cada senoide/ciclo é de 20 ms, logo, sabendo que

cada amostra demora aproximadamente 0,169ms (a partir de ensaios), tem-se que cada ciclo terá um total de 118-119 amostras. Como a precisão e exatidão dos valores obtidos aumentam com o número de amostras por ciclo, estabeleceu-se que seriam necessárias 10 vezes mais amostras que as inicialmente calculadas, ou seja 1185.

$$y[n] = 0,996 \times y[n-1] + 0,996 \times x[n] - 0,996 \times x[n-1] \quad (13)$$

Após a definição do ciclo são lidos os valores no ADC do Arduino, aos quais será aplicado o filtro digital de passa alto da equação (Eq. 13) para remoção do *offset* de 2,5V originado pelo divisor de tensão, permitindo que os valores possam oscilar novamente em torno de zero (Atmel, 2007). Durante o mesmo ciclo, partindo da fórmula (3) onde a corrente eficaz corresponde ao método RMS, é calculada a soma dos valores quadrados das amostras. Assim que termina o fim do ciclo é completado o método RMS com o cálculo da média e da raiz, sendo o valor obtido convertido para valores de corrente, através de:

$$I_{fator_de_conversão} = \frac{N_{espiras}}{R_{Resistência_de_Carga}} \times V_{resolução_ADC} \times I_{calibração} \quad (14)$$

O valor de I_{cal} representa um fator de correção dos valores obtidos com o sistema desenvolvido em comparação com a tomada de medição. Devido aos limites de resolução do ADC do Arduino, é necessário estabelecer como valor mínimo de medição de corrente, 0,3 A, sendo qualquer valor abaixo deste igual a 0.

Tendo o valor de corrente eficaz calculado, obtém-se o valor da potência aparente através da equação (4), que por sua vez utilizando a fórmula (5) permite determinar a potência ativa. Por último é possível o cálculo da energia através da expressão (6), utilizando a variação de tempo entre cada função loop() realizada, sendo o valor cumulativo com os anteriores. Importa referir que ambos os logaritmos da presença e do consumo energético estão condicionados ao envio de valores pela porta série, e registo de valores no cartão micro SD, de 1 em 1 minuto.

4.5.3 Código Processing

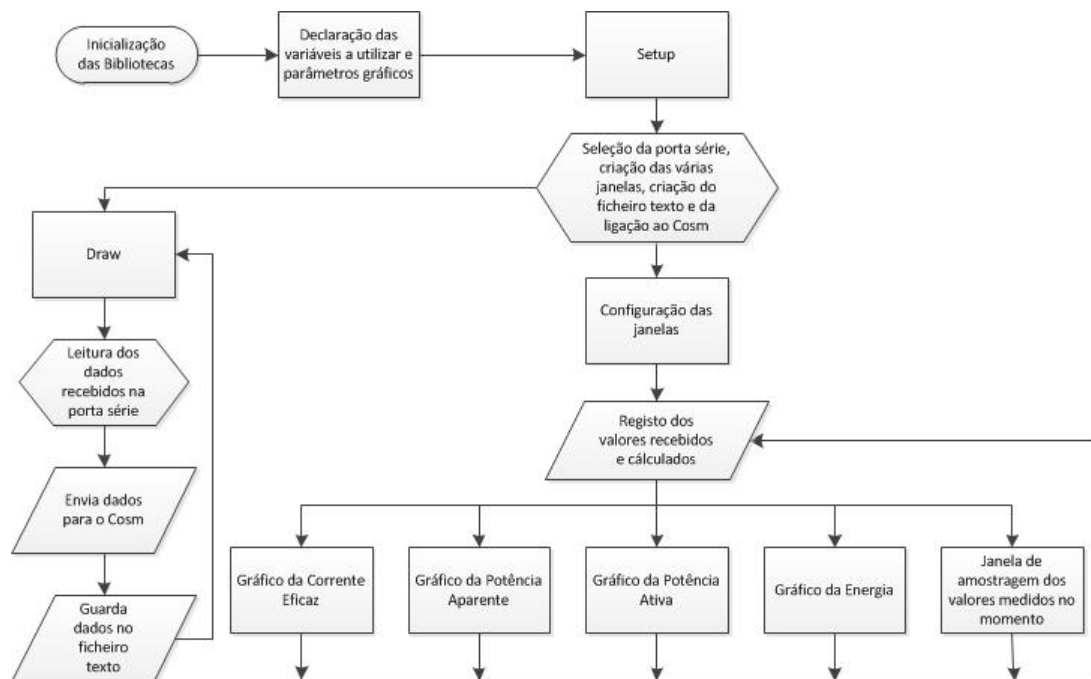


Fig. 19 - Fluxograma do sketch utilizado na programação do Processing

O código construído para o Processing tem, como já se referiu, a funcionalidade da amostragem numa interface dos dados comunicados pelo Arduino. Não obstante, devido à elevada versatilidade deste software, o Processing serviu também para a criação de um ficheiro .csv (*Comma-separated values*) para guardar os dados, e para o envio dos dados para a página de internet Cosm.

O fluxograma da Fig. 19 representa de forma resumida a estrutura do código desenvolvido para o conjunto de funções mencionadas anteriormente (ver Anexo C.). Caso tudo esteja configurado de forma correta, e o Arduino esteja a enviar os valores desejados, corre-se o programa, abrindo este em alguns segundos a interface desenvolvida, começando a guardar os dados no ficheiro .csv e a enviá-los para a *feed* selecionada na página Cosm. A partir deste momento os dados serão atualizados a cada minuto decorrido.

Como se pode ver pela Fig. 20, a interface desenvolvida em Processing é constituída por duas partes principais.

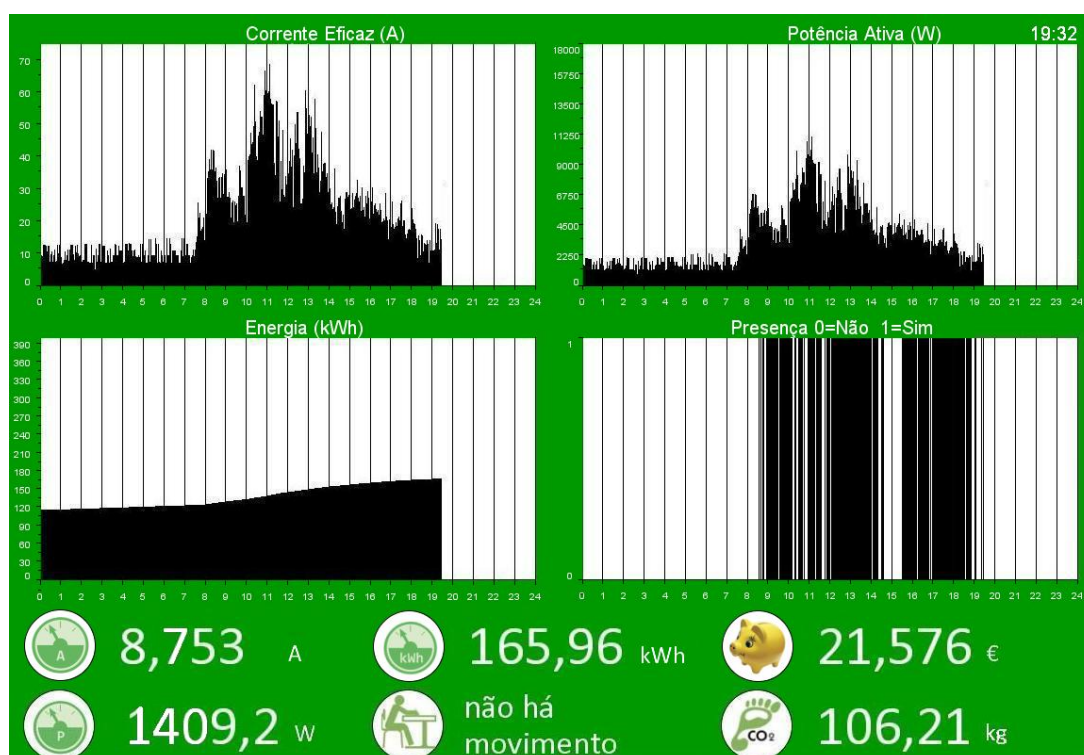


Fig. 20 - Interface de monitorização do sistema desenvolvido

Numa delas são representadas quatro janelas onde são produzidos os gráficos de quatro parâmetros analisados: corrente eficaz, potência ativa, energia e presença humana, todos em ordem ao tempo. Para cada um destes parâmetros é escolhida uma escala de valores para o eixo do Y que permita identificar as variações dos valores, enquanto no eixo do X se estabeleceu como um intervalo de valores para um dia de medição. É também possível a leitura das horas e minutos a que está a ser realizada a monitorização, sendo utilizados para o efeito os valores do RTC.

A segunda parte da interface indica os seis valores apresentados na parte inferior da janela para os parâmetros apresentados nos gráficos anteriores, além do custo e dióxido de carbono, correspondentes ao consumo de energia elétrica efetuado até ao momento. Os valores da corrente eficaz, potência ativa e presença humana são instantâneos, enquanto os de energia, dinheiro e dióxido de carbono são cumulativos. As atualizações de todos os valores são feitas de 1 em 1 minuto, consoante foi estipulado no código do Arduino. Em relação aos valores monetários, o preço da eletricidade para edifícios como os da faculdade não é tão linear e simples como o preço da eletricidade para uma habitação. Assim, assumindo os vários custos incluídos numa fatura da faculdade, assumiu-se 0,08 €/kWh como o preço aproximado da eletricidade. Por outro lado, o valor de dióxido de carbono corresponde à quantidade deste gás que é emitida por cada kWh de eletricidade produzido, para o caso Português, de 0.47 kg CO₂/kWh (Diário da República, 2008).

4.5.4 Interface Cosm

Tal como foi referido o Cosm terá a função de disponibilizar os dados enviados no Processing para a Internet, sendo indispensável para todo este processo a utilização da biblioteca “Cosm” no código Processing. Através da interface gráfica do Cosm é possível observar a variação dos dados da monitorização em qualquer local onde se esteja conectado à Internet. Para que tal aconteça é necessário associar à página de dados criada, o endereço IP da rede à qual está ligado o computador. Terminado este processo, será atribuído à página de dados um endereço único, que permitirá a qualquer pessoa aceder à mesma.

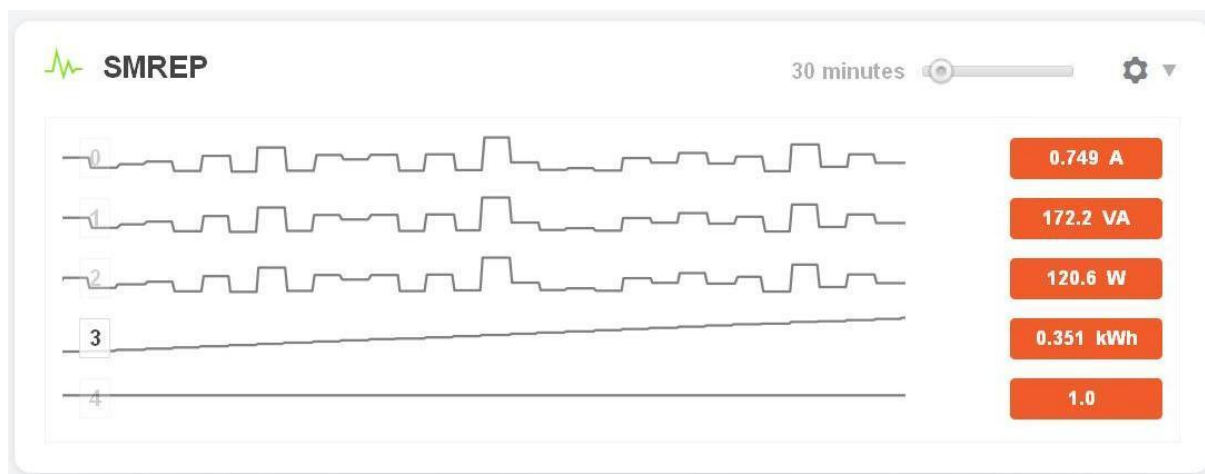


Fig. 21 - Interface produzido pelo Cosm

Dependendo da quantidade de valores que se enviam, como se observa pela Fig. 21 é possível comparar todos em simultâneo, sendo possível alterar o período de tempo de visualização, além de existir a possibilidade de uma visualização individual mais pormenorizada. Considerando as características da rede de internet da faculdade e o facto de criar endereços de IP privados, não foi possível criar uma página de apresentação no Cosm utilizando esta rede. Apesar disso, ficou demonstrado o funcionamento do Cosm, através da utilização de uma rede de internet doméstica.

4.6 Calibração do Sistema

Para que o sistema desenvolvido possa ser utilizado futuramente para medições de consumo de energia elétrica e deteção da presença de pessoas, foi primeiramente necessário calibrar o sistema. Através de um conjunto de vários ensaios efetuados ao TC e aos vários sensores do sistema de presença, foi possível verificar a precisão que o sistema poderia oferecer, sendo possível corrigir erros cometidos.

Após o sistema estar montado foram realizados uma série de ensaios teste, para comparação dos resultados obtidos com um medidor de referência, de forma a poderem ser estabelecidas as calibrações finais. Além disso, após esta calibração foram realizados ensaios sem e com Xbee, que permitiram uma primeira avaliação dos resultados obtidos durante um período de monitorização

4.6.1 Curvas de corrente medidas pelo TC

No decorrer da dissertação foram realizados uma série de testes para validação dos valores medidos pelo TC, sendo um dos principais a curva das correntes registadas no enrolamento primário. Neste ensaio foi utilizado o osciloscópio para comparação dos valores obtidos. Como foi referido, a corrente chega a nossas casas sob a forma de corrente alternada, pelo que ela nunca é constante, dando origem a curvas diferentes, consoante as características dos equipamentos que a estão a consumir. Estas características prendem-se com o tipo de cargas existentes, algo que foi bastante visível nas curvas obtidas. Como foi explicado no sistema desenvolvido, no caso do código implementado são utilizados os valores das amostras produzidas pelo Arduino, correspondentes aos valores de corrente medidos instantaneamente, para efeitos de cálculo da corrente eficaz verificada num dado intervalo de

amostragem. Devido à relação existente entre amostras e corrente alternada medida, foi necessário estudar e confirmar a semelhança de ambos os valores. Tendo em consideração as características do ensaio realizado foi necessário desenvolver um código para o Arduino capaz de receber uma elevada quantidade de dados num curto espaço de tempo (cada ciclo de 425 amostras realizado em 50 ms).

Fazendo ensaios em separado para um computador portátil, candeeiro com lâmpada incandescente, ferro de engomar, secador de cabelo, e um ensaio em simultâneo para o computador portátil e candeeiro, foi possível observar a semelhança entre as curvas produzidas pelas amostras registadas no Arduino e as curvas obtidas no osciloscópio.

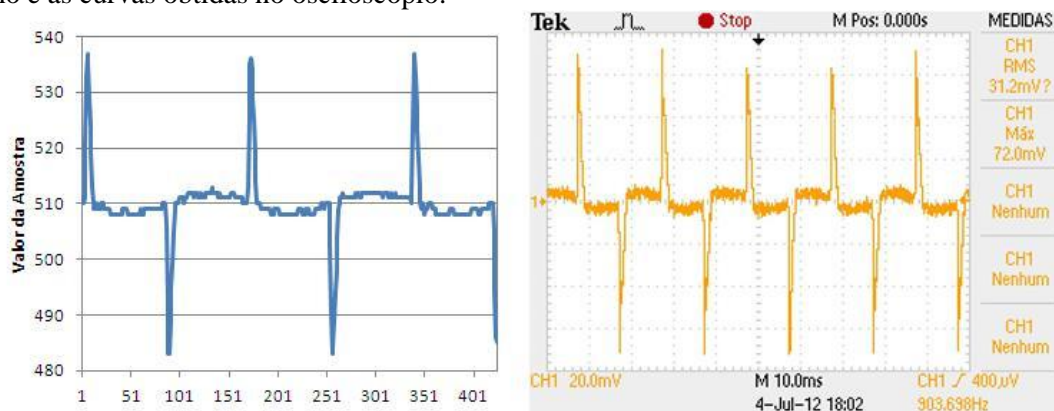


Fig. 22- Gráficos representativos da corrente consumida pelo portátil, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita)

No caso do portátil verificam-se, na Fig. 22, variações na amplitude das curvas em ambos os casos, sendo evidente um pico bastante característico dos equipamentos reativos e não lineares. Por outro lado, também se verificou que em ambos os casos o período era o mesmo, pois no caso da curva obtida pelos valores produzidos pelo ADC o tempo de um ciclo de amostragem foi de 50 ms, observando-se o mesmo tempo no osciloscópio. Por último, a parte da curva com amplitude mínima, que oscila aproximadamente junto do valor 510 (no caso do Arduino), tem para ambos os gráficos um ruído mínimo associado.

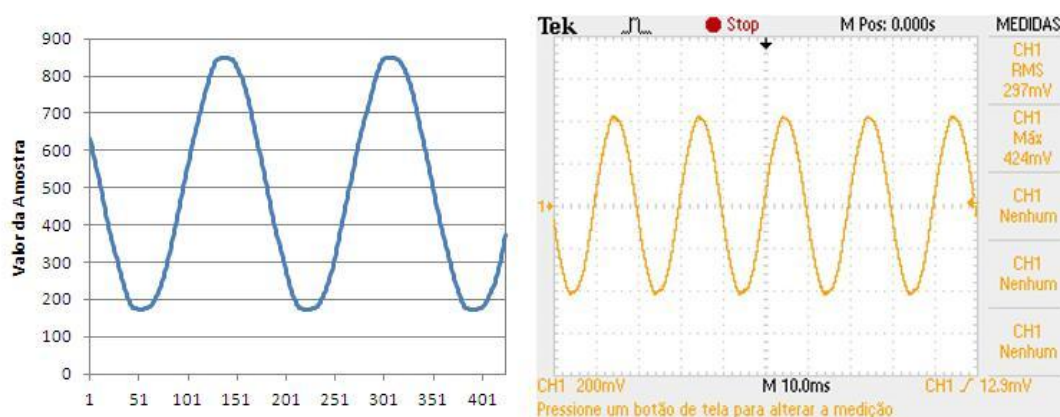


Fig. 23- Gráficos representativos da corrente consumida pelo ferro de engomar, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita)

No que concerne às curvas para o ferro de engomar (Fig. 24), ambas são representativas de equipamentos com características puramente resistivas, pelo que seriam de esperar curvas sinusoidais. Assim, como se pode observar pelo gráfico, a curva tem uma forma sinusoidal, sendo de realçar a qualidade da mesma, resultado de uma boa relação sinal/ruído. Devido ao elevado consumo deste género de equipamentos, a amplitude da onda é bastante maior, quando comparada com a do computador portátil.

Na curva do candeeiro, apesar de este ser um equipamento puramente resistivo (devido à utilização de uma lâmpada incandescente), observa-se um comportamento da curva diferente da do ferro de engomar num determinado aspeto (Fig. 24).

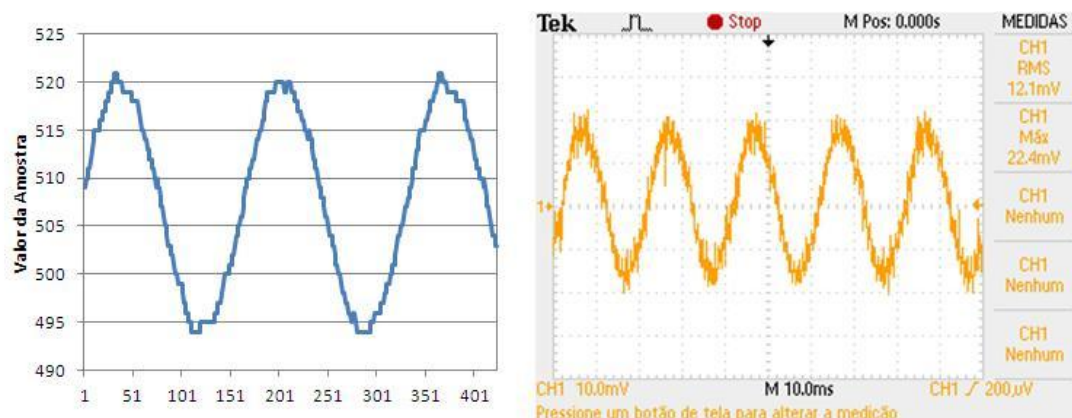


Fig. 24 - Gráficos representativos da corrente consumida pelo candeeiro, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita)

O traçado da curva não é tão perfeito como o da curva do ferro de engomar, revelando que as variações dos valores de corrente medidas estão próximos do limite da resolução do próprio ADC. Mais uma vez os dois gráficos produzidos de forma diferente assemelham-se, validando os dados obtidos através do Arduino.

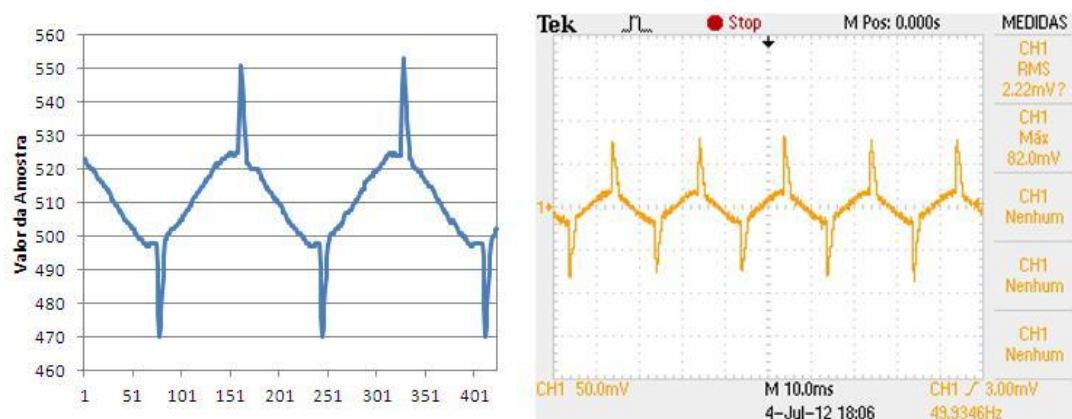


Fig. 25 - Gráficos representativos da corrente consumida pelo portátil mais o candeeiro, para valores das amostras do Arduino (à esquerda) e do osciloscópio (à direita)

Por último, em relação ao gráfico representativo do consumo efetuado por um candeeiro e um computador portátil a funcionar em simultâneo, observa-se que há uma soma das curvas produzidas por cada equipamento. Observando as amplitudes originais produzidas por cada aparelho constata-se, na Fig. 25, que para as amplitudes mais baixas foi utilizado o gráfico do candeeiro, enquanto os valores pico observados, correspondem à variação do consumo no computador portátil,

4.6.2 Tensão produzida pelo TC com resistência de carga

Após a comparação da forma das curvas obtidas para cada tipo de equipamento, importa verificar se os valores de corrente obtidos, através do transformador de corrente, seguem a curva de linearidade para a resistência de carga selecionada. Desta forma, realizaram-se vários ensaios utilizando um ferro de engomar por ter um alto consumo de corrente e ser uma carga puramente resistiva. Utilizando o valor de potência do equipamento, registado na tomada de monitorização de consumos⁴ e através da tensão e número de espiras do enrolamento secundário, foi possível calcular a corrente no enrolamento secundário. Através desta corrente, e sabendo o valor da resistência de carga, foi possível o cálculo da

⁴ Tomada de monitorização de consumo energético é um aparelho comercial, da marca Silvester, que indica valores instantâneos de corrente, tensão, fator potência, consumo de energia e custo.

tensão produzida à saída dos terminais da resistência. Importa referir que todo este cálculo tem como princípio a mesma expressão (9) implementada no algoritmo no Arduino, para conversão da corrente em tensão. Realizando vários ensaios, através da simulação do consumo do equipamento, gerado pelo número de voltas do cabo da fase ou neutro da extensão no interior do transformador, foi possível obter uma reta de valores de tensão para as várias correntes medidas e outra a partir dos valores de tensão medidos num osciloscópio (ver Fig. 26). Através deste passo foram gerados valores de corrente até 23A por um único equipamento, sendo resultado da multiplicação do número de voltas no transformador pelo consumo do ferro de engomar.

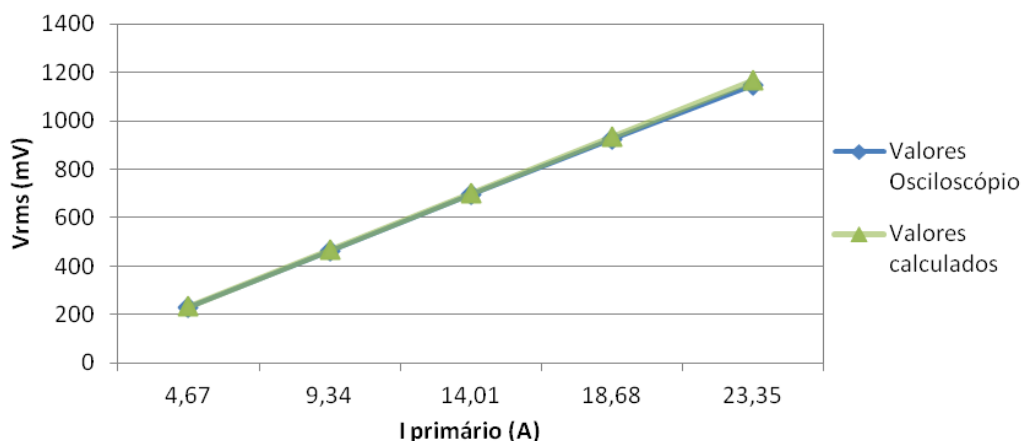


Fig. 26- Gráfico da tensão produzida para vários valores de corrente no primário ($R_{\text{carga}}=180\Omega$)

Em relação à linearidade dos valores, é possível chegar à conclusão de que até aos 23A, o transformador, com a resistência de carga escolhida, apresenta uma reta bem definida, tal como seria desejado, não evidenciando quaisquer sinais de saturação. Contudo apesar de um valor de corrente de 23A ser bastante aceitável para o que se mediu nos testes finais, seria desejável um ensaio para valores próximos do limite de corrente indicado pelo fabricante (75A), de forma a testar ao máximo o ponto de saturação da resistência escolhida. Neste caso devido ao limite do número de voltas dos cabos e do equipamento, tal não foi possível. Por outro lado comparando as tensões resultantes dos cálculos efetuados a partir da corrente medida no primário e das tensões medidas no osciloscópio é possível concluir que a diferença de valores é quase nula, pelo que se pode assumir que os cálculos efetuados estão corretos.

4.6.3 Verificação dos resultados produzidos pelos sensores IVP e ultrassons

De forma a ver a exatidão dos resultados obtidos através dos sensores de IVP e de ultrassons, durante aproximadamente 20 minutos, foi colocado numa sala escura o protótipo desenvolvido, a debitar os valores através da ligação pela porta série a um computador. Através de um código bastante semelhante ao desenvolvido na dissertação, mas apenas com informação relativa aos sensores de deteção de presença foi possível verificar em pormenor os resultados por estes registados. Das 44 medições feitas de 30 em 30 segundos verifica-se que em 8 delas os sensores IVP detetaram movimento, o que neste caso não corresponde ao esperado, pois não houve movimento de pessoas. Contudo, este erro pode ser justificado pela sensibilidade deste género de sensores a fatores como correntes de ar provenientes de uma janela aberta ou movimento de ar quente resultante da ventilação do computador. Além destas possíveis causas do erro obtido, em relação a uma sala normal outras ocorrências de erros podem justificar-se com as variações de radiação difusa provenientes do vidro de uma janela, ou da própria iluminação de uma sala. De certo modo poderia pensar-se que o tamanho dos cabos que ligam os sensores IVP ao Arduino poderia interferir com os dados obtidos. Contudo, através de vários ensaios realizados utilizando cabo curto e longo conclui-se que os resultados eram semelhantes e não poderia ser apontada como uma das causas para os erros destes sensores.

Em relação ao sensor de ultrassons verifica-se que através da introdução no código de uma variação mínima das diferenças entre distâncias obtidas (15cm), este não origina qualquer tipo de erro no valor de presença final.

4.6.4 Calibração da Corrente Eficaz

Apesar dos testes ao transformador de corrente revelarem um bom funcionamento do mesmo, foi necessário calibrar os valores de corrente calculados pelo Arduino com os da tomada de medição, sendo este definido como dispositivo de referência. Deste modo, foram realizados vários ensaios para cada um dos equipamentos disponíveis, que permitiram diminuir o desvio dos valores de corrente registados. A observação dos restantes parâmetros, passíveis de serem comparados, estabeleceram o tipo de erros que se podem encontrar, tendo em conta as especificações técnicas do sistema proposto.

Tabela 1 - Parâmetros monitorizados para vários equipamentos analisados

	Sistema Desenvolvido			Medidor SilverCrest				Erro Corrente (%)	Erro Potência Ativa (%)
	Corrente Eficaz (A)	Potência Aparente (VA)	Potência Ativa (W)	Corrente Eficaz (A)	Tensão (V)	Potência Ativa (W)	Fator Potência		
Monitor de Computador	0,443	101,8	71,3	0,443	248	62,8	0,57	0,0	-13,5
Computador Portátil	0,304	69,9	48,9	0,3	248	36,7	0,51	-1,3	-33,2
Secador	2,725	626,8	438,8	2,814	244	683,1	1	3,2	35,8
Ferro Engomar	4,914	1130,3	791,2	5,029	241	1211,9	1	2,3	34,7
Candeeiro com Lâmpada Incandescente	0	0	0	0,183	248	45,3	1	100,0	100,0
Secador+Monitor+Portátil	3,156	725,8	508,1	3,25	241	767,5	0,98	2,9	33,8
Candeeiro+Monitor+Portátil	0,817	187,9	131,5	0,817	245	136,1	0,68	0,0	3,4

Através de uma análise à Tabela 1, chega-se à conclusão de que os vários valores medidos variam de acordo com as características dos equipamentos monitorizados quer sejam resistivos, indutivos ou capacitivos. Contudo, esta relação não afeta a medição da corrente eficaz, que tem valores bastante semelhantes para cada equipamento, à exceção do candeeiro. Neste caso particular, o valor de 0A medido é justificado pelo facto do código implementado no Arduino não medir correntes inferiores a 0,3A, devido a serem valores fora do limite de resolução do ADC, ou seja, 0,049V.

Por um lado, em relação aos restantes parâmetros analisados, tem-se que equipamentos com cargas não puramente resistivas, como o monitor e o computador, têm um fator de potência relativamente baixo próximo dos 0,5-0,6. Este valor resulta numa potência ativa inferior à aparente que, quando comparada com as medições do sistema proposto, sofrem um desvio de aproximadamente 10W, resultando num erro de 13,5% e 33,2% para o monitor e o portátil, respetivamente. Por outro lado, o facto de se assumir que o fator potência tem o valor constante de 0,7 é prejudicial para a monitorização de equipamentos puramente resistivos como o secador e o ferro de engomar (com fator de potência 1), introduzindo uma elevada percentagem de erro ($\approx 35\%$).

Apesar destes aspetos negativos para as análises individuais aos equipamentos, caso se intersem vários tipos de equipamentos com cargas distintas, os resultados podem melhorar bastante. No ensaio realizado com o candeeiro, o monitor e o portátil em funcionamento simultâneo, o erro para a potência ativa foi bastante reduzido (3,4%), provando que esta mescla de cargas utilizadas, com valores de corrente próximos, contribui para que o fator potência se aproxime do estipulado no algoritmo do protótipo (0,7). Por outro lado se um dos equipamentos em funcionamento, como por exemplo no ensaio do secador, com o monitor e o portátil, tenha um consumo de corrente bastante superior aos

restantes (secador com 2,7A dos 3,2A totais consumidos), o aspeto vantajoso da mistura de cargas anteriormente referido é anulado.

4.6.5 Teste ao Sistema Proposto sem Zigbee

O primeiro teste, relativamente às medições disponibilizadas pelo Arduino para um período de monitorização, foi feito utilizando o protótipo sem a comunicação por Zigbee. Os ensaios foram realizados em dois espaços da faculdade, um gabinete e um laboratório. No gabinete o objetivo foi medir o consumo gerado por um reduzido número de equipamentos, através do cabo fase ou neutro de uma extensão, e a variação da presença humana, permitindo os resultados chegar a uma primeira avaliação da evolução e qualidade dos dados produzidos ao longo do tempo.

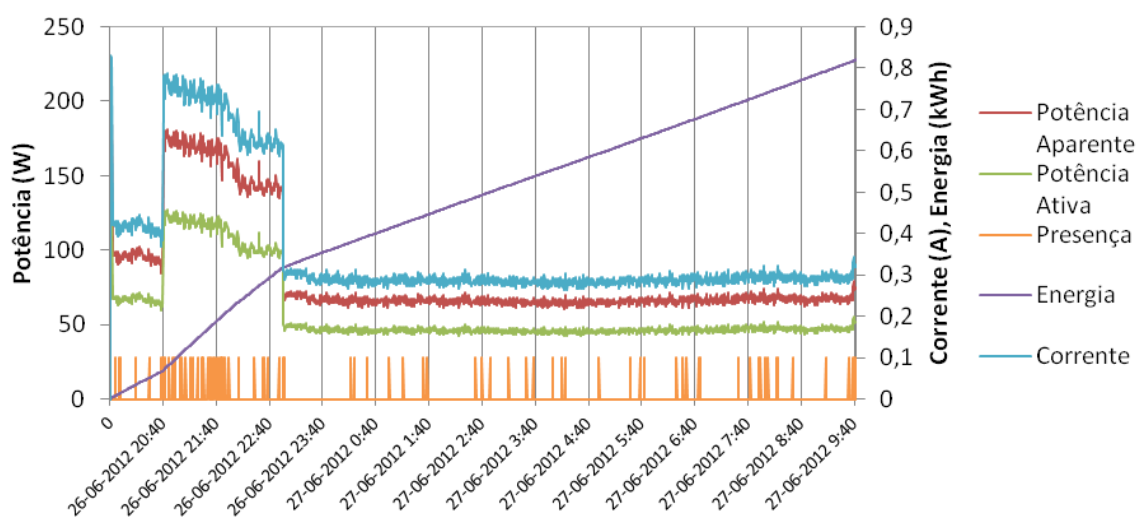


Fig. 27 - Gráfico da variação do consumo de energia elétrica e presença no gabinete

Assim, através de uma monitorização durante 14 horas, concentrada no período noturno, observa-se pelo gráfico da Fig. 27 uma diferença notória dos consumos quando há presença de pessoas no gabinete. Das 20h30 até às 23h verifica-se que a corrente atingiu valores próximos dos 0,7A (portátil mais monitor em funcionamento), enquanto no restante período o valor da corrente anda próximo de 0,3A (apenas portátil em funcionamento). Como resultado direto deste aumento, constata-se um crescimento da potência aparente e ativa para aproximadamente 160W e 120W, respetivamente, causado pelos valores de tensão eficaz e de fator potência constantes que permitem o seu cálculo. Da mesma forma, durante este período de ocupação o declive da reta do consumo de energia segue os maiores valores de corrente consumidos, tendo um aumento destacável face ao restante período de tempo. Se se analisar a variação da presença pelos sensores, o resultado não é de todo o esperado. Apesar de se notar uma maior concentração de presença durante o período de maior consumo, o que está correto, durante a noite, mesmo não havendo qualquer entrada de pessoas na sala, os sensores detetaram o movimento de pessoas. Com este facto confirmam-se estar de acordo os erros observados no ensaio da sala escura para verificação de erros dos sensores de IVP.

A medição efetuada no laboratório foi feita a partir da instalação do protótipo no quadro elétrico da sala, enquanto os sensores de deteção foram colocados à entrada da porta. Sabendo das limitações do sistema relativamente a instalações monofásicas, devido ao facto de neste caso se estar perante um quadro trifásico, optou-se por selecionar através de uma pinça amperimétrica a fase com valores de corrente mais elevados num dado momento. Comparando com o ensaio no gabinete, variou-se o período de monitorização para quatro dias, incluindo dois de fim de semana, com o objetivo de verificar o comportamento dos sensores de presença.

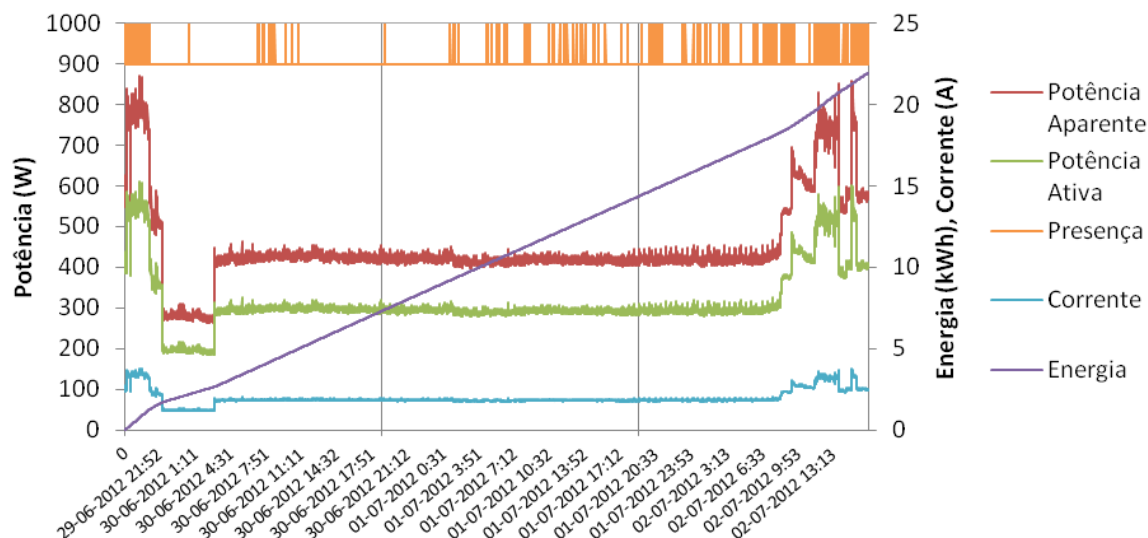


Fig. 28 - Gráfico da variação do consumo de energia elétrica e presença no laboratório

Tal como foi verificado na monitorização anterior, também nesta a variação do consumo de energia elétrica foi mais conclusiva do que a própria variação da presença. Pela análise da variação da presença, conclui-se que é extremamente difícil chegar a qualquer tipo de relação lógica, pois inúmeras vezes foi detetado movimento de corpos para períodos de tempo em que o mesmo não é expetável. Comparando os restantes valores da parte do consumo, e tendo em consideração os resultados do teste para o gabinete, verifica-se que para esta fase os valores de corrente variaram aproximadamente entre 1- 4A, o que se traduz em maiores valores das potências ativa e aparente. Como seria de esperar, estes valores de correntes superiores encontram-se nos dias úteis (29 de junho e 2 de julho), enquanto no fim de semana o valor permanece na ordem de 1,85A, correspondendo a um maior número de equipamentos em funcionamento.

4.6.6 Teste ao protótipo com utilização do Zigbee

Os dois ensaios analisados anteriormente utilizaram a ligação USB entre o Arduino e o computador para comunicação dos dados contudo, como já foi mencionado, a comunicação sem fios é um dos principais objetivos do sistema desenvolvido. Desta forma foi testada a comunicação Zigbee segundo o esquema da Fig. 4, para o caso do teste no gabinete. Após várias tentativas constatou-se que os dados que estavam a chegar à porta série do computador, a partir do Xbee coordenador, continham uma série de erros. Estes erros puderam ser confirmados de duas formas, através do terminal do XCTU (utilizado na configuração do Xbee) e do ficheiro de dados criado pela interface Processing.

Tabela 2 - Exemplo dos erros presentes nos valores obtidos utilizando o Zigbee

Data	Corrente	Potência Aparente	Potência Ativa	Energia	Presença
05-07-2012 11:17	0,258	59,2	41,5	0,687	1
05/07/121 20:00:00,266	61,2	42,9	0,689	5	7
05-07-2012 11:21	0,288	66,2	46,3	0,6	1

Observando a Tabela 2 é possível constatar os problemas da utilização do Zigbee em modo transparente para o envio de tantos dados em tão pouco tempo. A dessincronização que ocorreu diversas vezes implicou que determinados valores fossem entendidos como outros parâmetros, alterando o conteúdo final dos dados em vários momentos. Num dos ensaios teste realizado, dos 850 dados registados apenas 260 não apresentavam qualquer deformação o que corresponde a um erro de 70%.

5. Aplicação do Sistema Proposto

Um dos propósitos finais, após a construção do protótipo e realização dos vários testes de calibração, foi a monitorização durante o período de uma semana, de um espaço da faculdade. Deste modo, foi escolhido o bar do C1 por ter expetável movimento de pessoas e constantes variações do consumo elétrico ao longo de um dia. Com o intuito de desenvolver uma solução final mais próxima daquilo que é um protótipo, com ligações entre sensores e Arduino, sem recurso à placa de ensaio, soldaram-se os componentes e ligações necessárias ao funcionamento de cada sistema, utilizando a *shield* para esse fim. Além disso, ambos os sistemas foram colocados em caixas para uma maior proteção.

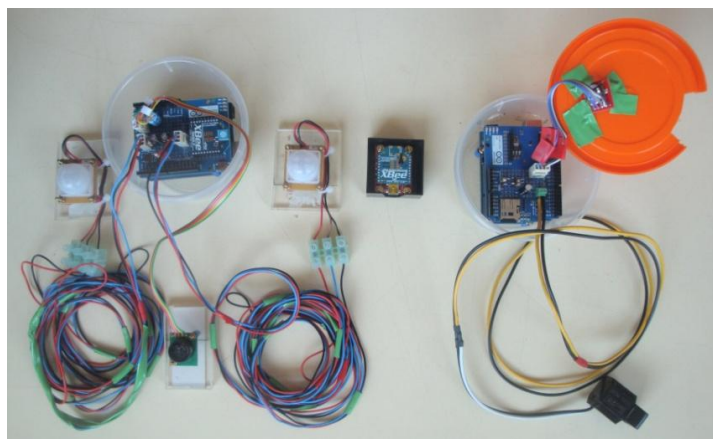


Fig. 29 - Montagem final dos dois sistemas a instalar no bar do C1

Devido à distância entre o quadro elétrico e o próprio bar foi necessário pensar numa solução que permitisse realizar ao mesmo tempo a monitorização da presença de pessoas e do consumo elétrico, sem que a mesma colocasse em causa a segurança das medições e do equipamento (Fig. 29). A solução encontrada foi separar os dois sistemas utilizando dois Arduinos, sendo que no caso da presença, se colocou um computador ligado por USB ao coordenador Xbee na Associação de Estudantes, recebendo os dados provenientes do sistema instalado no bar. Este sistema foi alimentado por um conversor AC/DC compatível com o Arduino. Em relação à monitorização da energia elétrica, tal como no laboratório, também neste caso se procedeu à escolha de uma das fases a monitorizar pelo sistema, sendo os dados enviados por USB para um computador ou guardados no cartão micro SD.

Tabela 3 – Características técnicas de cada sistema

	Número de amostras por ciclo	Intervalo de medição dos <i>clamps</i> de corrente (A)
Protótipo	1185	0 - 75*
CA8332	256	10 - 6500
*Nota: Arduino só mede $I > 0,3A$		

Além disso, para esta medição contou-se com a instalação de um aparelho comercial indicado para medições deste género, um CA 8332 da Chauvin Arnoux igual ao descrito nos sistemas existentes, tornando assim possível a comparação dos valores obtidos. A Tabela 3 indica as características de cada sistema, sendo importante notar que as mesmas podem contribuir para a obtenção de resultados diferentes.

Tabela 4 – Parâmetros medidos por cada sistema

	Corrente Eficaz	Tensão Eficaz	Fator Potência	Potência Aparente	Potência Ativa	Energia Elétrica	Presença
Protótipo	X			X	X	X	X
CA8332	X	X	X	X	X	X	

No caso particular da monitorização efetuada, cada equipamento é capaz de medir um conjunto de parâmetros, sendo alguns deles apenas obtidos num ou noutro sistema, tendo em consideração as características técnicas de cada um (Tabela 4).

A monitorização realizada começou no dia 27 de julho às 15 horas e terminou no dia 3 de agosto às 14 horas e 59 minutos, perfazendo uma semana de medições.

Tabela 5 - Horário de funcionamento do bar do C1

Dias	Chegada das Funcionárias	Abertura do Bar	Fecho do Bar	Saída das Funcionárias	Horário de Ocupação
27, 30 e 31 julho	7h30	8h00	19h00	19h15	7h30 – 19h15
1, 2 e 3 agosto	8h00	8h30	18h00	18h15	8h00 – 18h15
28 e 29 de julho	Não funciona por ser fim de semana				

Apesar de se monitorizar a presença humana, tendo em conta as possíveis limitações do sistema, bem como o facto de estar restringido a apenas uma zona do bar, importa mencionar o horário de funcionamento do espaço para a semana de monitorização (Tabela 5). Para efeitos de análise dos consumos considera-se como horário de ocupação o período desde a chegada das funcionárias até à sua saída.

5.1 Medições e Resultados

De seguida são apresentados os gráficos obtidos para os vários parâmetros analisados, sendo a cor verde representativa dos valores obtidos recorrendo ao protótipo, enquanto os de cor azul mostram os valores obtidos a partir do CA8332. Os resultados são disponibilizados na sua grande maioria para o período de tempo total da monitorização.

5.1.1 Corrente Eficaz

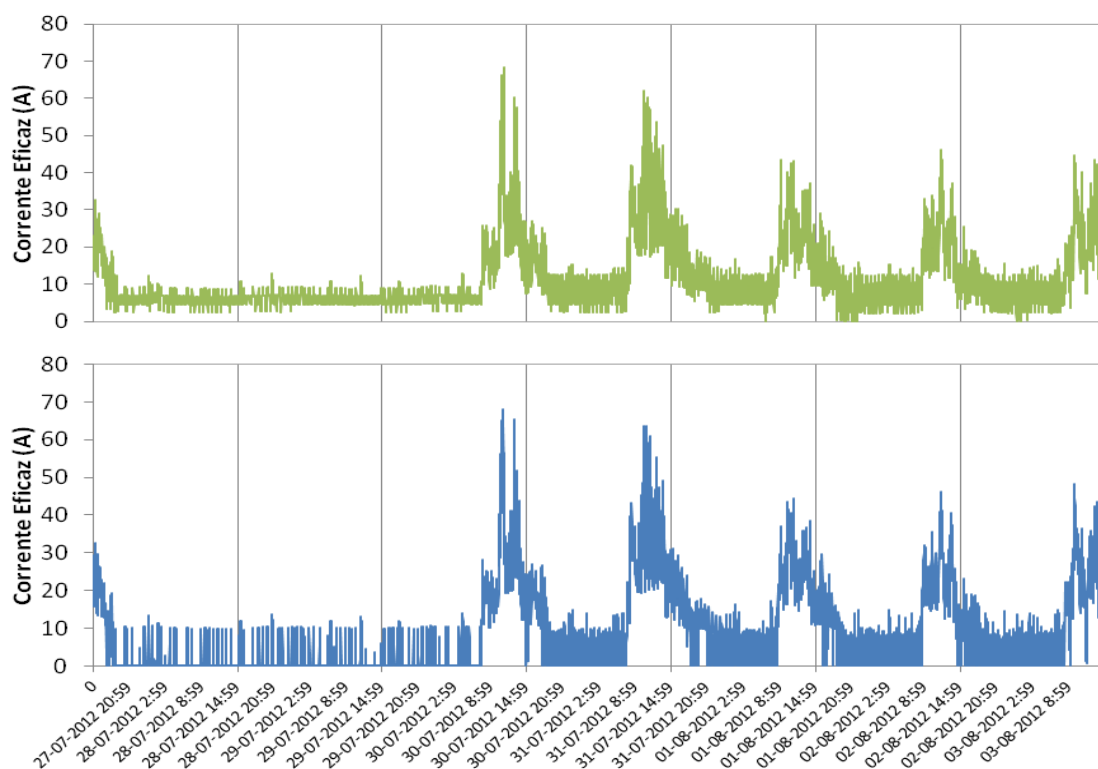


Fig. 30 - Gráfico de comparação da corrente eficaz para o período total de monitorização (resultados do protótipo a verde e resultados do CA8332 a azul)

Uma primeira análise aos gráficos de corrente eficaz (Fig. 30) permite identificar os padrões de consumo do espaço monitorizado, verificando-se que os maiores valores de corrente correspondem ao horário de ocupação.

A partir do mesmo gráfico, é possível identificar o momento em que as funcionárias entram no bar, a abertura do bar, o momento de preparação do almoço, o período em que são servidos os almoços e o restante tempo de funcionamento do bar até à hora de saída das empregadas. Como se pode observar existem dois padrões distintos de base de consumo, um para dias úteis e outro para dias de fim-de-semana (28 e 29 de julho). Apesar de o gráfico permitir identificar a semelhança da evolução das curvas, tanto para o horário de ocupação como para sem ocupação, nos cinco dias úteis e no fim de semana, revela-se importante uma análise mais detalhada para cada um destes três períodos. Assim, são escolhidos períodos de tempo de monitorização bastante mais curtos, capazes de permitirem uma melhor caracterização do consumo elétrico efetuado.

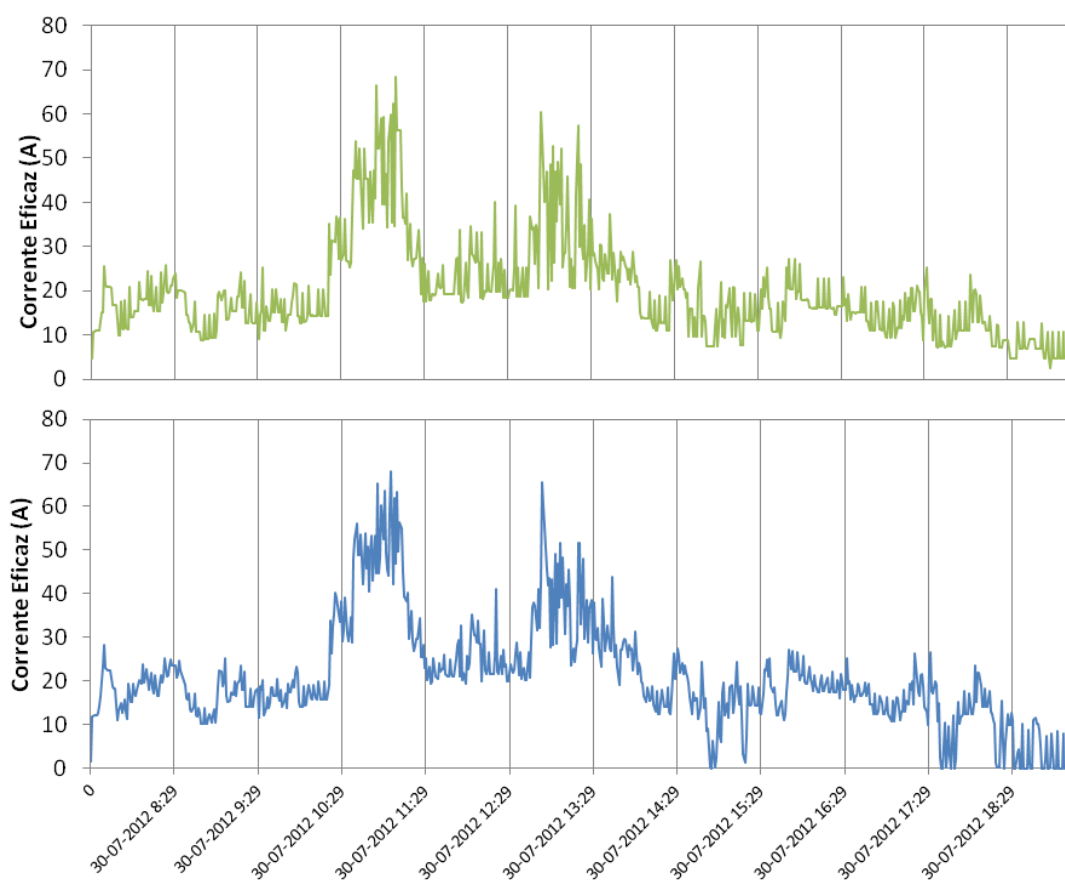


Fig. 31 - Gráfico de comparação da corrente eficaz para o horário com ocupação num dia útil (resultados do protótipo a verde e resultados do CA8332 a azul)

Para a evolução do consumo durante o horário de ocupação foi escolhido o dia de 30 de julho (Fig. 31). Pelas 7h30 da manhã verifica-se um aumento da corrente, correspondendo à entrada das funcionárias no bar. Até à abertura do bar aos clientes, a corrente oscila entre os 10A e os 25A, podendo estes valores ser causados pela ligação da iluminação, máquina do café, termoacumulador e ar condicionado. A partir das 9 horas o bar é aberto, voltando a aumentar o valor da corrente para os 20-25A. Após a abertura, o valor da corrente oscila de acordo com a utilização de equipamentos como a torradeira, tostadeira, micro-ondas e máquina do café, nunca ultrapassando os 25A. No intervalo de tempo entre as 10h e as 11h30 regista-se um pico de corrente próximo dos 70 A, neste caso justificado pela confeção do almoço na cozinha. Como se pode ver pela Fig. 30, o valor deste pico varia bastante de dia para dia da semana, pois depende do tipo de refeição servida, que por sua vez influencia o tipo de equipamentos utilizados (ex.: grelhador, forno elétrico, exaustor, etc). Após a preparação do almoço observa-se uma diminuição da corrente entre os 20 e os 40A. Próximo das 13h o valor da corrente volta a subir atingindo picos de consumo entre os 50-60A, correspondentes ao período em que se

servem as refeições de almoço. Este pico tem como possíveis causas, por exemplo, o funcionamento das máquinas de lavar e secar a loiça, a máquina do café, entre outros. No período após as 14h e até à saída das funcionárias, o consumo volta a descer, atingindo valores próximos dos observados para o início da manhã, podendo ser originados pelos equipamentos já mencionados (10-30A).

Importa mencionar que os gráficos, produzidos por cada um dos sistemas de monitorização do consumo de eletricidade, evidenciam as mesmas características da curva e um quase perfeito sincronismo. Porém, algumas vezes os valores máximos e mínimos da corrente têm pequenas diferenças quanto ao valor registado num e noutro gráfico ou, então, observam-se picos que aparecem em apenas um dos gráficos. Uma das possíveis causas para esta diferença de valores resulta das medições não estarem totalmente sincronizadas.

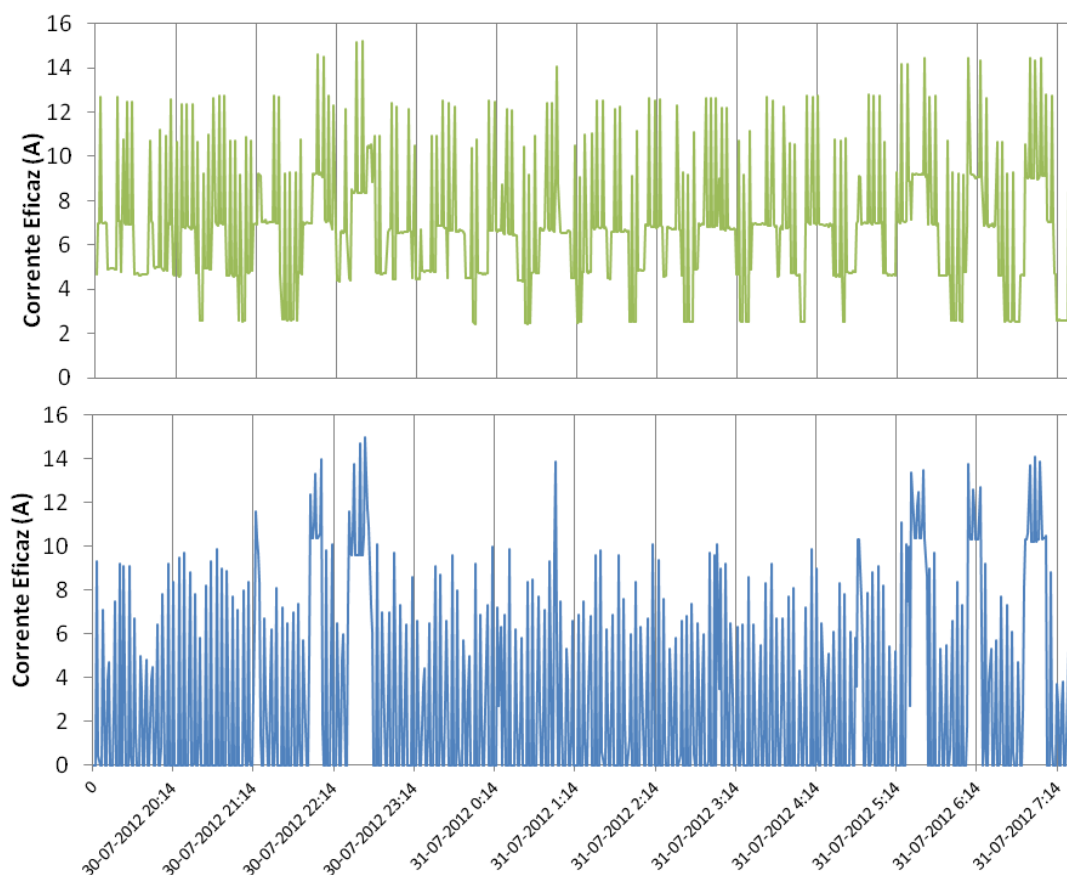


Fig. 32 - Gráfico de comparação da corrente eficaz para o horário sem ocupação num dia útil (resultados do protótipo a verde e resultados do CA8332 a azul)

Continuando a análise para o dia 30 de julho no caso da evolução da corrente consumida para horário sem ocupação, os valores devem-se aos equipamentos que permanecem ligados durante este período, incluindo todas as máquinas frigoríficas existentes no bar (à exceção de duas vitrinas frigoríficas), a máquina do café e o termoacumulador. Uma comparação entre os dois gráficos da Fig. 32 revela que tanto o comportamento como os valores obtidos são diferentes para o sistema desenvolvido e para o CA 8332. Destacam-se principalmente dois pontos de distinção nos gráficos, o limite máximo e mínimo de cada gráfico. Em relação ao protótipo, o limite máximo normalmente obtido aproxima-se de 11-12,5A, enquanto no caso do sistema de referência o valor normalmente alcançado é de 9-10A. Por sua vez no limite mínimo, enquanto no sistema proposto é na ordem de 2,5A sendo alcançado num máximo de quatro vezes por hora, para o CA 8332 ele é de 0A e é obtido continuamente durante o período sem ocupação. Esta discrepância de resultados indica que para o protótipo existe um consumo base de 2,5A durante todo o período sem ocupação e para o sistema de referência existem períodos em que nenhum equipamento está a consumir energia. A justificação para o CA 8332 não registar valores

tão baixos de corrente prende-se com o intervalo de medição dos seus *clamps* (Tabela 3). Mesmo assim seria de esperar que qualquer valor abaixo de 10A não fosse medido, contudo observa-se no caso analisado valores entre os 4 e 10A. Como este é um problema do próprio equipamento não foi importante para efeitos da tese.

Outra das particularidades dos gráficos é de que ao longo do tempo se constata que os valores estão em contínua alternância, e de que os picos obtidos não estão sincronizados em certos períodos. Além dos valores limites de cada gráfico, percecionam-se vários níveis de corrente bem definidos para cada análise. Tal facto pode ser justificado por parte dos equipamentos em atividade durante este período do dia, estarem permanentemente a ligar/desligar-se (efeito do termostato no funcionamento). Uma nota para os picos ocasionais que se verificam ao mesmo tempo em cada um dos gráficos, e que ultrapassam os limites anteriormente mencionados, atingindo correntes entre os 13A e os 15A (ex: 22h, 22h30, 1h e 5h30).

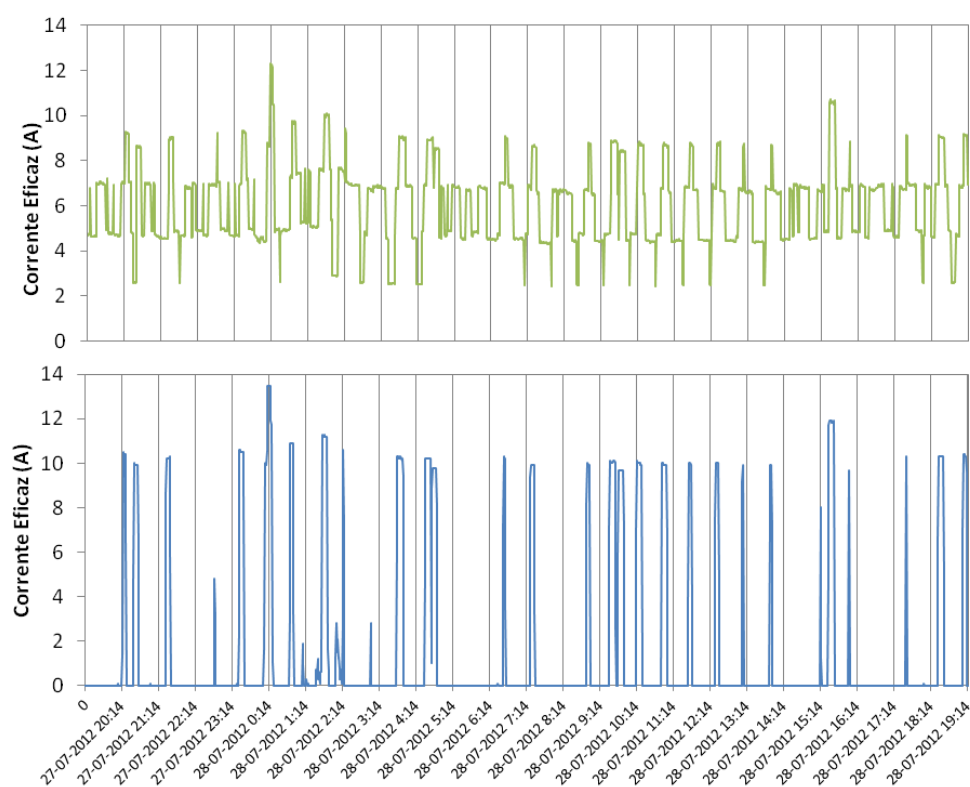


Fig. 33 - Gráfico de comparação da corrente eficaz num dia de fim de semana (resultados do protótipo a verde e resultados do CA8332 a azul)

No que respeita à evolução da corrente consumida num dia de fim de semana, sabendo que não há qualquer presença humana no bar, apenas resulta dos equipamentos que permanecem ligados durante este período, ou seja, todas as máquinas frigoríficas, à exceção de duas vitrinas frigoríficas. Pode-se concluir que à exceção da máquina do café e do termoacumulador o cenário de consumo é semelhante ao de um dia útil no horário sem ocupação.

É de notar, mais uma vez, a discrepância dos resultados obtidos nos gráficos dos dois sistemas de monitorização utilizados (Fig. 33), importando fazer uma análise para cada um dos cenários. No caso do sistema proposto constata-se que até à meia-noite de dia 28 de julho, o gráfico tem três níveis de corrente compostos por um mínimo de 2,6A, um intermédio de 4,7A e um máximo de 9A, sendo esta variação provocada pelo ligar e desligar das máquinas frigoríficas. Os períodos de tempo em que os níveis de corrente se mantêm constantes podem ir desde poucos segundos até à dezena de minutos, dependendo da máquina frigorífica que se ligue/desligue. O facto de nunca serem atingidos os 0A é mais uma vez justificado por todos os equipamentos nunca se desligarem em simultâneo. Por volta das 20h45 horas destaca-se um pico de 13A, de origem não identificada. A partir da meia-noite passam a existir apenas dois dos três níveis de corrente anteriores, o de 4,7 e 9A.

Em relação ao segundo gráfico, como foi referido, o comportamento é bastante diferente, não existindo três níveis distintos ao longo da monitorização, mas apenas um de aproximadamente 10A. Tal como aconteceu no horário sem ocupação para um dia útil, também neste caso o problema de não serem praticamente medidos valores abaixo dos 10A, se justifica pelo intervalo de medição dos próprios *clamps*. De referir que o mesmo pico de 13A identificado pelo protótipo, também foi medido pelo CA 8332 tendo um máximo de 13,6A.

Comparando, os gráficos da Fig. 32 e da Fig. 33 depreende-se que os valores de consumo estão em contínua alternância para curtos períodos de tempo. No primeiro e no segundo caso os mesmos podem permanecer constantes durante minutos ou horas, conforme o cenário estudado. Sabendo que a única diferença nos dois cenários são o funcionamento da máquina do café e do termoacumulador apenas nos dias úteis, conclui-se que eles são responsáveis pela variação contínua da corrente que impede que haja uma nivelção tão explícita como no cenário de fim de semana.

5.1.2 Tensão Eficaz

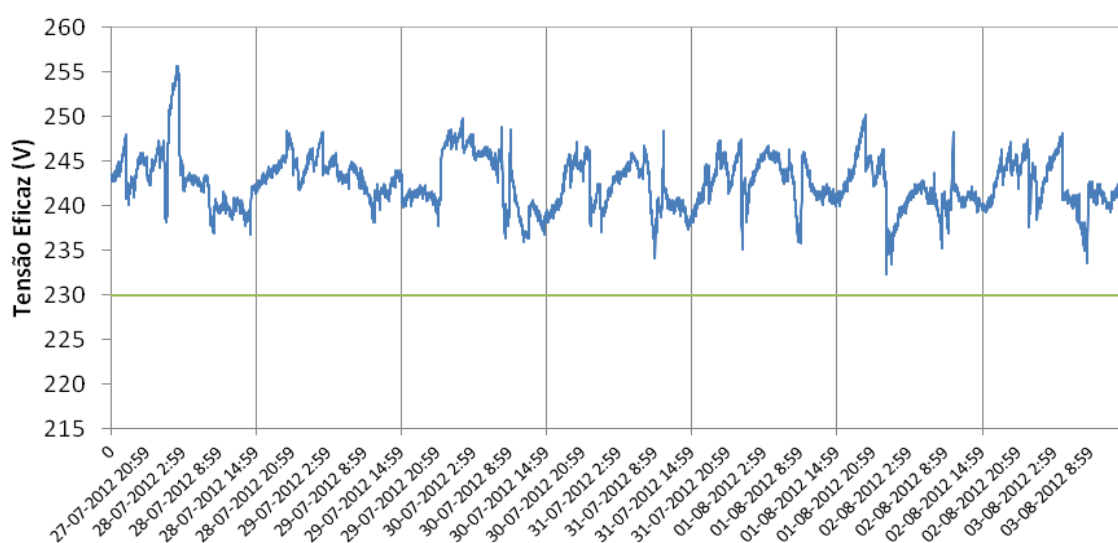


Fig. 34 - Gráfico de comparação da tensão eficaz (resultados do protótipo a verde e resultados do CA8332 a azul)

Foi mencionado, na descrição do sistema proposto que o mesmo não iria medir o valor de tensão instantânea, assumindo-se um valor constante de 230V ao longo do tempo. Contudo, aproveitando o facto do aparelho de referência medir a tensão, é possível obter o gráfico da evolução semanal da tensão eficaz (Fig. 34).

Conclui-se que o valor da tensão nunca é constante e que para o bar está acima dos 230V estipulados, sendo a média semanal de 243V. Assim, o erro introduzido pela não medição da tensão é aproximadamente de 5,3% para o caso estudado. Importa salientar que este valor pode variar de acordo com o espaço/edifício monitorizado e as condições de fornecimento de eletricidade a que está sujeito. Um dos fatores que pode justificar um valor superior ao referido pela EDP é a existência de um posto de transformação no edifício do bar.

5.1.3 Fator Potência

Por não ser medida a tensão, também não pode ser calculado o fator de potência, assumindo-se neste caso um valor constante (0,7). Mais uma vez tirando partido das capacidades do CA 8332, entre elas, o cálculo de valores instantâneos de fator potência, é de todo proveitoso a comparação com o protótipo, embora não seja um dos objetivos. Como se pode ver pelo gráfico (Fig. 35), os valores encontrados para cada período de tempo estão de acordo com os equipamentos em funcionamento, verificando-se um padrão de fator de potência consoante o período do dia que se monitoriza.

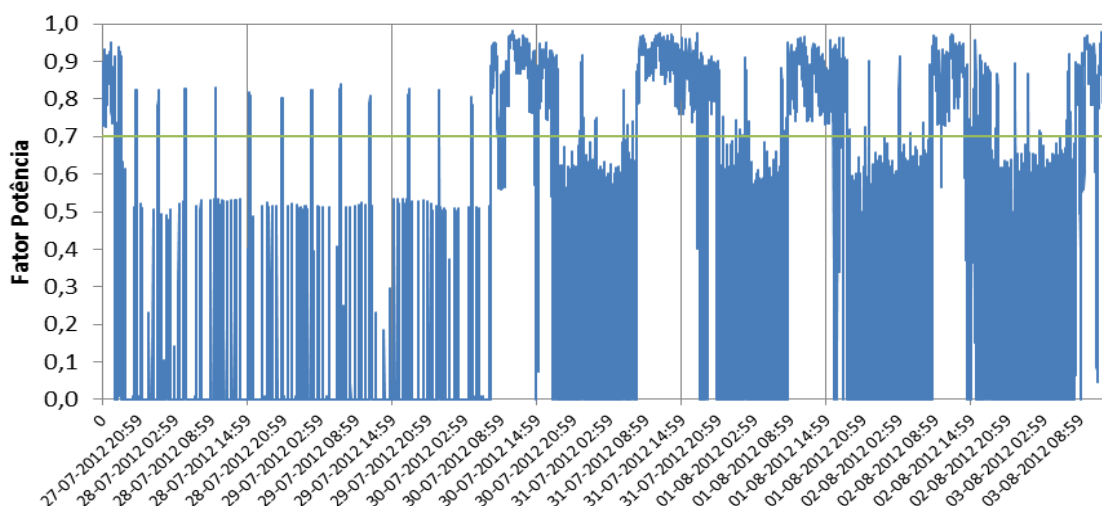


Fig. 35- Gráfico de comparação do fator potência (resultados do protótipo a verde e resultados do CA8332 a azul)

No caso do fim de semana o fator potência oscila entre os 0,5 e os 0, sendo resultado do consumo efetuado apenas pelas máquinas frigoríficas. Partindo da fórmula do fator de potência, sempre que a corrente medida é zero, o fator de potência também o é, correspondendo nos gráficos ao momento em que uma e outra situação acontecem. É de notar que aproximadamente de 6 em 6 horas o fator de potência atinge os 0,8. Sabendo que um frigorífico convencional doméstico tem um fator de potência próximo de 0,8, conclui-se que quando se observa este valor ele deve estar associado unicamente ao consumo dos frigoríficos convencionais existentes no bar enquanto os 0,5 a vários equipamentos em simultâneo incluindo as arcas congeladoras. Devido à impossibilidade da análise individual de cada equipamento é difícil concluir que equipamentos são responsáveis por cada fator potência.

Tomando em consideração os dias úteis constata-se que, também neste caso, existe uma relação entre o valor do fator de potência e o período em análise. Durante o horário de ocupação, o fator de potência situa-se entre 0,8 e 1, justificado pela utilização de cargas do tipo resistiva (fornos e grelhadores). Após a saída das funcionárias, aproxima-se dos valores de fim de semana, embora tenha uma variação mais acentuada, equiparada à da corrente. Tendo em conta todos os valores medidos pelo CA8332 durante a semana de monitorização, pode ser calculada uma média do fator potência, que no caso é de 0,67, valor bem próximo do assumido no protótipo.

5.1.4 Potência Aparente

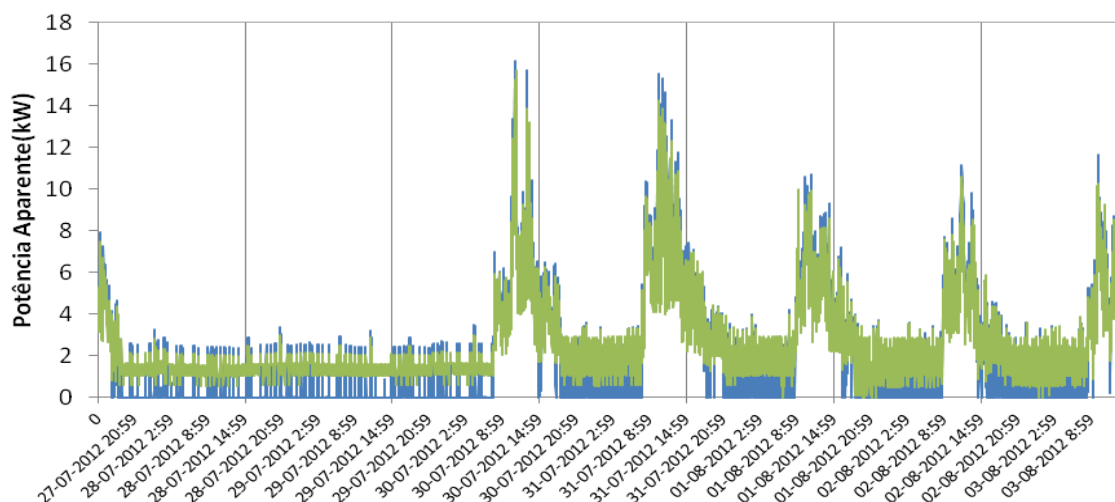


Fig. 36 - Gráfico de comparação da potência aparente consumida (resultados do protótipo a verde e resultados do CA8332 a azul)

A representação da potência aparente e ativa medida de ambos os medidores tem como objetivo a percepção dos valores de potência alcançados. No caso da potência aparente, o gráfico da Fig. 36 segue diretamente o padrão de variação do gráfico da corrente eficaz (Fig. 30), tendo valores superiores aos da potência ativa. A diferença dos valores obtidos para cada medidor é igual à que se verifica para o gráfico da corrente.

5.1.5 Potência Ativa

No caso dos valores de potência ativa, como estes são obtidos a partir do fator potência, as diferenças entre os sistemas são maiores (Fig. 37).

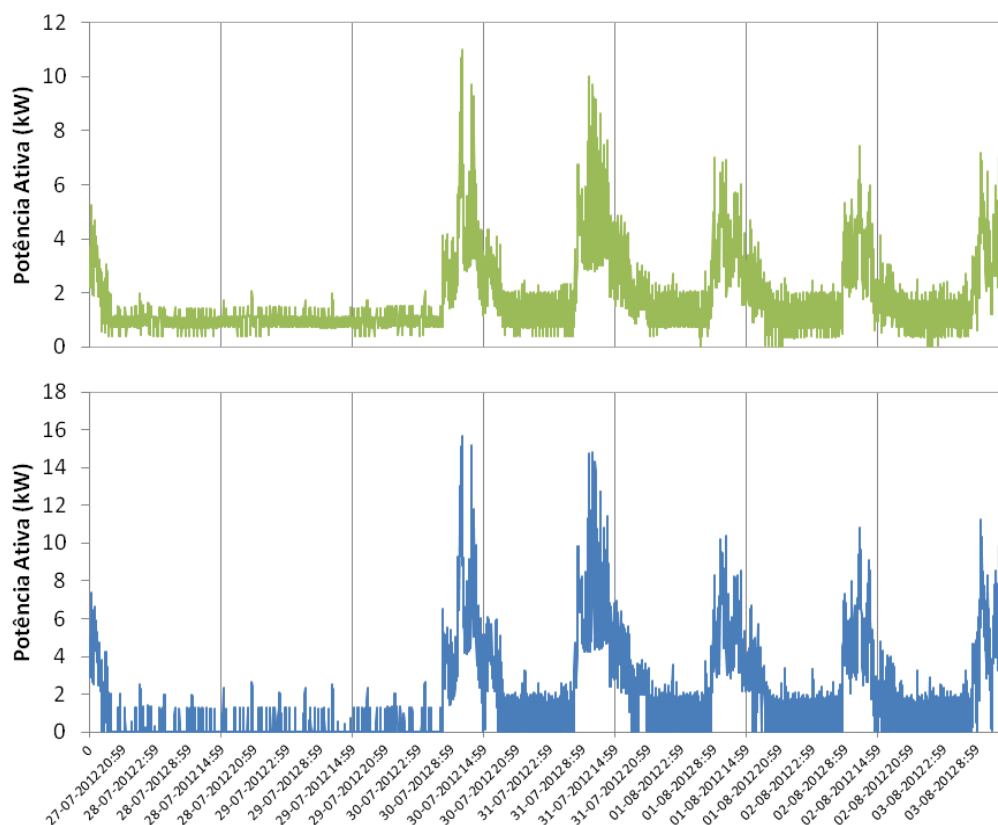


Fig. 37 - Gráfico de comparação da potência ativa consumida

Por exemplo, nos consumos dentro do horário de ocupação verifica-se uma grande diferença que atinge em algumas situações aproximadamente 7kW (hora de almoço). Este desvio deve-se em grande medida à diferença do fator potência para este período, 0,7 para o protótipo e 0,8-1 para o CA8332. Em relação aos restantes períodos de análise, a diferença de potências medidas oscila em torno de 1kW, sendo este desvio justificado pelo CA8332 não medir abaixo dos 10A. De notar que quando a corrente é superior aos 10A para estes períodos sem ocupação, o valor de potência é semelhante ao do protótipo.

5.1.6 Energia Elétrica

O gráfico da Fig. 38 traduz a evolução do consumo de eletricidade ao longo da semana. Como a energia é apenas um resultado do cálculo das variáveis anteriormente analisadas, esta varia de acordo com os gráficos anteriormente examinados, sendo que, para períodos de maiores valores de corrente/potência a inclinação do gráfico é maior e vice-versa. Assim, observa-se que no fim de semana o protótipo regista um aumento linear do consumo de energia de 50 kWh, enquanto para o CA8332 o aumento é bastante mais reduzido em torno de 10 kWh. Esta diferença de valores justifica-se pelos diferentes valores de corrente medidos para este período. Em relação ao período de ocupação, a variação do consumo também é distinta para cada caso. Por exemplo, para o dia 30 de julho o

sistema desenvolvido calcula um aumento da energia de 37 kWh, ao passo que o sistema de referência calcula um aumento de 52 kWh. No caso do horário sem ocupação para um dia útil observa-se um aumento superior do consumo para o protótipo na ordem dos 14 kWh, para o dia 30/31 de julho, face a um valor próximo dos 7 kWh para o CA8332.

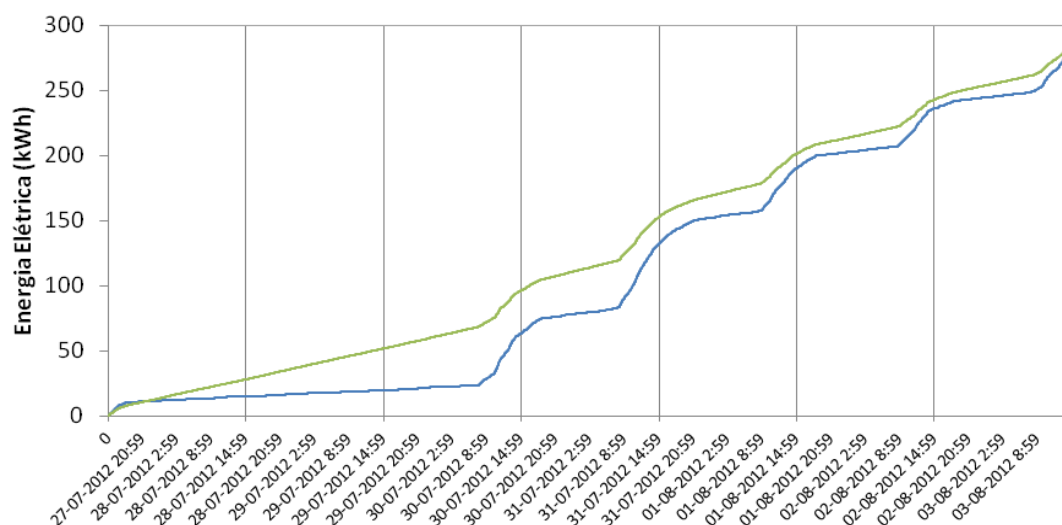


Fig. 38 - Gráfico de comparação do consumo de energia elétrica (resultados do protótipo a verde e resultados do CA8332 a azul)

A evolução das duas curvas permite notar que ao longo da semana de medição a diferença dos consumos vai diminuindo, através de um balanceamento entre um maior consumo observado pelo protótipo para o fim de semana e, por sua vez, um maior consumo observado pelo CA8332 para os períodos no horário de expediente.

Tabela 6 - Comparação dos valores de energia consumida, custo da energia e CO₂ emitido

	Energia (kWh)	Custo (€)	CO ₂ (kg)
Protótipo	284	22,7	181
CA8332	280	22,4	179

Prova disso, é o facto de que através dos valores finais de energia consumida para a semana monitorizada (Tabela 6), é possível chegar a um erro de 1,4% entre a energia indicada pelo sistema desenvolvido e a energia indicada pelo sistema comercial. Comparando este erro com o da potência ativa, pode parecer estranho uma discrepância tão acentuada mas, no caso da energia, tem-se um valor cumulativo que de acordo com o já explicado tem um erro baixo devido à anulação das diferenças entre os valores obtidos, à medida que se avança na semana. Por seu turno o erro da potência é uma média dos erros de todas as potências instantâneas, incluindo o erro produzido pelo CA8332 para o período do fim de semana.

Tendo em consideração os valores de consumo obtidos, importa aferir um enquadramento com o estabelecido no RSECE para valores limite dos consumos globais dos edifícios de serviços existentes. No caso particular de restaurantes o IEE (indicador de eficiência energética) é de 170 kgep/m².ano (RSECE, 2006). Logo, sabendo que o fator de conversão para a eletricidade é de 0,29 kgep/kWh (RCCTE, 2006), que o bar tem 220m² e um ano tem aproximadamente 52 semanas, calcula-se um consumo semanal de 2480 kWh. Assim conclui-se que no caso do bar os consumos para a semana de monitorização estão bastante abaixo do limitado por lei, podendo ser apontada como uma das principais causas a reduzida potência de climatização instalada, tendo em consideração as necessidades do espaço.

Partindo dos valores do consumo de eletricidade e recorrendo aos fatores de conversão para o preço e emissões de CO₂ por kWh, já mencionados no interface Processing, calcula-se que o consumo de

eletricidade no bar resulte num custo próximo de 23€ e num total de emissões de 180 kg de CO₂ para a semana monitorizada (Tabela 6).

5.1.7 Presença

Apesar de os dados do consumo traduzirem a atividade de pessoas no bar, tirando partido das capacidades do sistema de monitorização da presença, obteve-se o gráfico representativo da ocupação do mesmo espaço (Fig. 39).

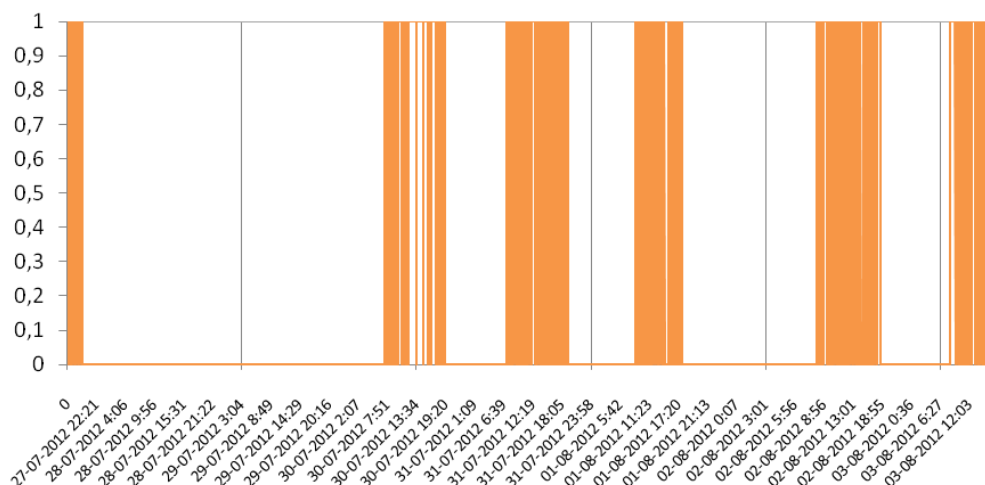


Fig. 39- Gráfico indicativo da presença humana no bar

Embora tenham ocorrido alguns problemas ao nível da comunicação por Zigbee e avaria dos sensores de IVP, foi possível obter dados demonstrativos de que são detetadas pessoas quase permanentemente no bar, desde a entrada até à saída das funcionárias. Como já foi referido é durante este período de deteção que se verificam maiores consumos de eletricidade. Seria de esperar uma relação para certos períodos do dia, como o lanche, de que o facto do sensor ultrassons estar colocado na zona de atendimento, pudesse justificar eventuais aumentos de consumo. Contudo a permanência da funcionária nesta área e os problemas de comunicação impossibilitaram uma análise mais aprofundada.

Tabela 7 - Valores da primeira e última deteção para cada dia da monitorização

Dias	Primeiro Deteção	Última Deteção
27 de julho	-	19h33
30 de julho	7h14	19h13
31 de julho	7h18	19h17
1 de agosto	8h34	18h00
2 de agosto	8h12	18h41
3 de agosto	8h18	-

Os dados obtidos para a primeira e última deteção em cada dia (Tabela 7), permitem estabelecer na maioria dos casos uma relação direta com o horário de ocupação do bar (Tabela 5). É possível observar que nos dias 1, 2 e 3 de agosto já se verifica o novo horário do bar, com duração de atividade mais curta, enquanto nos primeiros dias de monitorização a primeira e última deteção foram mais cedo e mais tarde, respetivamente. É de referir que durante os dias de fim de semana e o período noturno dos dias úteis não foi verificada qualquer deteção de presença, como seria de esperar, provando que o sensor ultrassons não produz qualquer falha a este nível.

Por último, em relação ao problema da comunicação por Zigbee, para a semana analisada seriam de esperar um total de 10080 valores. Porém foram recebidos 7667 o que corresponde a uma omissão de dados de 24%.

6. Conclusões

O desenvolvimento do sistema de monitorização de energia elétrica e da presença humana foi realizado com sucesso. Tendo em consideração as várias características inicialmente estipuladas para o sistema, é de destacar o facto de ter sido possível monitorizar os vários parâmetros associados ao consumo de energia elétrica, e da presença humana num edifício. Por sua vez foi possível comunicar e armazenar os dados para um computador portátil e para um cartão micro SD, respetivamente. O desafio de criar uma interface capaz de mostrar os dados em tempo real foi alcançado com êxito, assim como a disponibilização dos dados para consulta na internet. Posteriormente os dados armazenados foram passíveis de serem utilizados para a construção de gráficos, em folhas de cálculo, para uma melhor interpretação dos mesmos.

Os vários ensaios realizados tendo em vista a calibração do sistema permitiram testar cada um dos sensores, concluindo sobre a qualidade dos resultados por eles produzidos, quando comparados com a realidade. Em relação ao transformador de corrente utilizado, tanto os valores de corrente como os de tensão produzidos à sua saída são bastante próximos do esperado. Apenas se salientam as limitações do sistema para valores baixos de corrente medida, devido à resolução do ADC do próprio Arduino. Por outro lado, a calibração dos sensores utilizados para o sistema de deteção de presença revelou, no caso dos sensores IVP, que estes não produzem os resultados esperados, uma vez que detetam movimento quando o mesmo não existe. Por seu turno, o sensor de ultrassons não demonstrou qualquer tipo de erros na deteção de movimentos, sendo compatível com a utilização no sistema proposto.

Após a fase de calibração dos sensores do sistema estar concluída foi possível realizar várias aplicações do sistema para medições teste. Estas monitorizações incluíram numa primeira fase análises individuais ao consumo energético de vários equipamentos e numa segunda o comportamento tanto a nível de consumo como de presença humana, em espaços da faculdade. Na grande maioria dos ensaios do consumo de eletricidade foram utilizados sistemas existentes no mercado para comparação dos resultados finais obtidos e para posterior validação dos mesmos.

No caso da análise individual, conclui-se que o facto de não ser medida a tensão limita o sistema para uma correta medição do consumo gerado por estes equipamentos, caso se afastem do fator de potência assumido (0,7). Por outro lado, quando se faz a medição de vários equipamentos em simultâneo, a percentagem do erro diminui de acordo com o tipo de equipamentos utilizados.

O principal teste realizado ao sistema foi a monitorização do bar do C1. As características físicas do próprio espaço implicaram que o sistema de monitorização fosse alterado. Desta forma o sistema teve de ser separado em dois, um para o consumo de energia e o outro para deteção de presença humana. Os resultados obtidos para a monitorização do consumo de eletricidade, tendo em conta as características do sistema, foram bastante positivos, uma vez que mediu sem qualquer interrupção todos os valores, sendo também possível observá-los na interface construída. Na maioria das vezes os valores de corrente corresponderam aos registados pelo equipamento de referência, excetuando os valores de corrente inferiores ao intervalo de medição deste equipamento. Em relação aos erros dos resultados, apesar de no caso da potência ativa serem evidentes desvios face ao esperado, conclui-se que no final não impedem que os valores de energia calculados estejam próximos dos registados pelo aparelho de referência.

No que respeita à deteção da presença humana, apesar do sistema ter sido testado várias vezes, tendo sempre funcionado, observou-se que os sensores IVP se danificaram não podendo detetar o movimento durante toda a monitorização. Assim, os resultados finais apenas dependeram dos valores registados pelo sensor de ultrassons, tendo como aspeto positivo o facto de não se observarem os erros produzidos pelos sensores IVP durante a calibração. Além deste problema, a utilização do Zigbee para comunicação da informação obtida, do Arduino para o computador portátil, implicou que alguns dos dados enviados fossem perdidos, devendo-se o problema à pouca solidez do modo de comunicação utilizado. Apesar disso, o gráfico obtido vai de encontro ao horário de ocupação do espaço monitorizado, sendo possível estabelecer uma relação entre períodos de ocupação e os maiores consumos de eletricidade.

Tendo em conta a influência que este tipo de monitorizações pode ter no comportamento do consumidor, é de esperar que a apresentação detalhada da eletricidade consumida, custos financeiros e ambientais associados, e ocupação de um determinado espaço, possa sensibilizar para alteração da sua forma de atuação.

Face a alguns aspetos menos positivos observados no protótipo final, estabelecem-se como pontos a melhorar:

- a medição da tensão na rede, permitindo desta forma um cálculo do fator instantâneo a cada momento, além de um valor de potência ativa mais preciso;
- implementação de um modo de comunicação mais eficiente e robusto, como por exemplo o recurso ao modo API compatível com os módulos Xbee utilizados;
- utilização de sensores IVP de maior qualidade que não produzam resultados errados.

Como balanço final do trabalho desenvolvido durante a tese, esta foi extremamente positiva, pois permitiu integrar vários conceitos físicos e matemáticos nas áreas da eletrónica e programação informática. Foi também possível integrar conceitos da comunicação de dispositivos e redes sem fios.

Por outro lado o sistema desenvolvido, tanto a nível de hardware como software utilizado, permitiu alcançar o conceito da Internet das Coisas e da Gestão de Energia.

7. Referências

- Acroname Robotics. 2011.** Devantech Sonic Range Finders (SRF). *Página Web da Acroname*. [Online] 2011. [Citação: 3 de junho de 2012.] <http://www.acroname.com>.
- Arduino. 2012.** Arduino. *Página Web do Arduino*. [Online] 2012. [Citação: 26 de maio de 2012.] <http://www.arduino.cc/>.
- . **2012.** Arduino Uno. *Página Web do Arduino*. [Online] 2012. [Citação: 26 de maio de 2012.] <http://arduino.cc/en/Main/ArduinoBoardUno>.
- . **2006.** SRFxx Sonic Range Finder Reader. *Página Web do Arduino*. [Online] 2006. [Citação: 19 de junho de 2012.] <http://arduino.cc/en/Tutorial/SFRRangerReader>.
- Atmel. 2007.** AVR:465: Single-Phase Power/Energy Meter with Tamper Detection. 2007.
- Bildr. 2011.** Did It Move? Detecting Motion with PIR + Arduino. *Página Web Bildr*. [Online] 2011. [Citação: 19 de junho de 2012.] bildr.org/2011/06/pir_arduino/.
- Bureau of Energy Efficiency.** Energy Monitoring and Targeting. *General Aspects of Energy Management & Energy Audit*.
- Chauvin Arnoux. 2004.** C.A 8332 C.A 8334. 2004.
- Comunidade Europeia. 2006.** Diretiva 2006/32/CE do Parlamento Europeu e do Conselho. *Eficiência na utilização final de energia e aos serviços energéticos*. 2006.
- CR Magnetics.** Folha de Produto CR 3100 Series. *Split Core Current Transformer-CR3100 Series*.
- Darby, Sarah. 2006.** The effectiveness of feedback on Energy consumption. s.l. : Environmental Change Institute-University of Oxford, 2006.
- Dent, Evan. 2010.** Energy Savings – a Strategy in Utilities. *Endress Hauser*. [Online] 2010. [Citação: 23 de julho de 2012.] <http://www.za.endress.com>.
- Diário da República. 2008.** 2.^a série — N.º 122. 2008.
- Doukas, Charalampos. 2012.** *Página Web da Building Internet of Things*. [Online] 2012. [Citação: 2012 de julho de 31.] <http://www.buildinginternetofthings.com/>.
- EDP Energias de Portugal. 2012.** *Página Web da EDP Serviço Universal*. [Online] 2012. [Citação: 23 de julho de 2012.] <http://www.edpsu.pt/>.
- Elkor Technologies Inc. 2006.** *AN0305 Introduction to Current Transformers*. Londres : s.n., 2006.
- Faludi, Robert. 2011.** *Building Wireless Sensor Networks*. s.l. : O'Reilly, 2011.
- Glolab. 2011.** How Infrared motion detector components work. *Página Web Glolab*. [Online] 2011. [Citação: 26 de abril de 2012.] <http://www.glolab.com/pirparts/infrared.html>.

IEA, International Energy Agency. 2009. *Key World Energy Statistics*. 2009.

IEEE. 2006. WIRELESS MAC AND PHY SPECIFICATIONS FOR LR-WPANS. *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. 2006.

Kranengurg, Rob van. 2008. *The Internet of Things: A critique of ambient technology and the all-seeing network of RFID*. s.l. : Institute of Network Cultures, 2008.

Lombard, Luis Pérez, Ortiz, José e Pout, Christine. 2007. A review on buildings energy consumption information. 2007.

Paiva, José. 2007. *Redes de Energia Elétrica: uma análise sistémica*. s.l. : IST Press, 2007.

Portugal Telecom. 2012. Meo Energy. *Página Web da PT*. [Online] 2012. [Citação: 2012 de julho de 25.] <http://www.lojapt.pt/Pages/ProductDetail.aspx?PRID=3109>.

Processing. 2012. *Página Web do Processing*. [Online] 2012. [Citação: 8 de agosto de 2012.] <http://processing.org>.

Rayond A. Serway, John W. Jewet Jr. 2004. *Physic for Scientists and Engineers with Modern Physics*. s.l. : Brooks/Cole-Thomson Learning, 2004.

RCCTE. 2006. DL 67/2006. 2006.

RSECE . 2006. DL 79/2006. 2006.

SAFARIC, S. e MALARIC, K. 2006. *Zigbee wireless standard*. s.l. : 48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications, 2006.

SESUL. 2011. Proposta de colaboração do SESUL no projecto Universidade Verde. 2011.

Springer. 2010. *Handbook of Modern Sensors*. s.l. : Springer, 2010.

Wikipedia. 2012. Arduino IDE. *Página Web da Wikipedia*. [Online] 2012. [Citação: 9 de agosto de 2012.] <http://en.wikipedia.org/wiki/Arduino#Software>.

—, 2012. Modelo OSI. *Página Web da Wikipedia*. [Online] 2012. [Citação: 30 de junho de 2012.] http://pt.wikipedia.org/wiki/Modelo_OSI.

—, 2012. Universal Serial Bus. *Página Web da Wikipedia*. [Online] 2012. [Citação: 2 de agosto de 2012.] http://pt.wikipedia.org/wiki/Universal_Serial_Bus.

Wiring. 2012. *Página do Wiring*. [Online] 2012. [Citação: 8 de agosto de 2012.] <http://wiring.org.co>.

ZigBee Alliance. 2012. ZigBee Alliance. *Página Web da ZigBee Alliance*. [Online] 2012. [Citação: 20 de junho de 2012.] <http://www.zigbee.org/>.

ZigBee Standards Organization. 2006. ZigBee Specification. 2006.

Anexo A: Preços do Material

Componente	Preço s/IVA (€)	Distribuidor
Arduino Uno Ver. 3	20	Inmotion
Arduino Wireless SD Shield	19,9	Inmotion
Transformador de Corrente CR Magnetics 3110	13,4	Digikey
2x Sensor Infravermelhos Passivo Hanse SE-10	17,5	Inmotion
Sensor Ultrassons Devantech SRF02	20	PTRobotics
Relógio Tempo Real Maxim DS1307	9,9	PTRobotics
Router Xbee Digi XB24-Z7CIT-004	29,5	Inmotion
Router Xbee Digi XB24-W7CIT-004	29,5	Inmotion
Sparkfun Breakout Board	19,95	Inmotion
Componentes Diversos	10	Vários
TOTAL	179,65	

Anexo B: Código de Programação do Arduino

```
#include <Wire.h> //I2C library
#include <SD.h> //SD library
#define DS1307_ADDRESS 0x68 //RTC library
byte zero = 0x00; //workaround for issue #527 (RTC variable)
//definition of pin on Arduino board
const int PIR1 = 2;
const int PIR2 = 3;
const int LED = 5;
const int pinPot = A0;
const int CS_pin = 4; //select pin for the SD card print data
const int pow_pin = 8; //selct pin for power supply
//Motion Detection Variables
int val1 = 0; //PIR1 value
int val2 = 0; //PIR2 value
int reading = 0; //US value
int oldreading = 0; //oldUS value
int count = 0; //counter to keep LED tu
int calibrationPIR = 0; //calibration time of PIR sensors
int presence_state = 0; //sets initial value of presence as 0, without presence
//Energy Consumption Variables
const int numberOfSamples = 1185;
//Callibration Constants
const int Vrms = 230; //Aparent power calculated using this Vrms
const float PF = 0.70; //set an average Power Factor value
//CR Magnetics CR31000-3000 CT parameters
const float CT_BURDEN_RESISTOR = 75;
const float CT_TURNS = 3000;
```

```
//Calibration coefficients
int calibrationCT = 0; //calibration time of CT sensors
const float ICAL = 0.96; //set the current calibration coefficient using the
//current measured on
// silvercrest 0,544A divided by current measured with clamp 0,581A
int lastSampleI,sampleI; //Sample variables
float lastFilteredI, filteredI; //Filter variables
float sqI, sumI; //Power calculation variables
float Irms; //Useful value variables
float apparentPower,realPower;
float kilowattHour;
unsigned long last_kwhTime;
unsigned long kwhTime;
//Real Time Clock Variables
int second, minute, hour, weekDay, day, month,year;

void setup(){
  Wire.begin(); // start the I2C bus
  Serial.begin(9600); // open the serial port:

  //Definition of input/output values
  pinMode(PIR1,INPUT);
  pinMode(PIR2,INPUT);
  pinMode(LED,OUTPUT);
  pinMode(pinPot,INPUT);
  pinMode(CS_pin, OUTPUT);
  pinMode(pow_pin, OUTPUT);
  digitalWrite(pow_pin, HIGH);
```

```
String header = "day-month-year hour:minute:second;Irms (A);Apparent Power (W);Real Power (W);Energy (kWh);Presence";

Serial.println(header);

//Initialize Card
if (!SD.begin(CS_pin))
{
    Serial.println("Card Failure");
    return;
}

File logFile = SD.open("SRMEP_SD.csv", FILE_WRITE);
if (logFile)
{
    logFile.println(header);
    logFile.close();
}
else
{
    Serial.println("Could not read command file.");
    return;
}

void loop(){
    runRTC();
    passiveIR();
    ultrasonic();
    presence();

    if(second == 0){ //condition to run functions only from 1 to 1 minute
        printDate();
```

```
consumption();

if( calibrationPIR < 1 ){ //set a time callibration to wait for the sensor to be
    // prepared and this condition avoid a first value of 1 on serial port
    presence_state=0;
    calibrationPIR++;
}
Serial.println(presence_state);

File logFile = SD.open("SRMEP_SD.csv" , FILE_WRITE);
if (logFile)
{
    //print to micro SD card
    logFile.print(day);
    logFile.print("/");
    logFile.print(month);
    logFile.print("/");
    logFile.print(year);
    logFile.print(" ");
    logFile.print(hour);
    logFile.print(":");
    logFile.print(minute);
    logFile.print(":");
    logFile.print(second);
    logFile.print(";");
    logFile.print(Irms,3);
    logFile.print(";");
    logFile.print(apparentPower,1);
    logFile.print(";");
    logFile.print(realPower,1);
    logFile.print(";");
```

```
logFile.print(kilowattHour,3);
logFile.print(";");
logFile.println(presence_state);
logFile.close();
}

presence_state=0;
}

delay (600); //set a delay to avoid the print on serail port of two lines of values
}

void passiveIR(){
  val1 = digitalRead(PIR1); //receives the PIR1 values
  val2 = digitalRead(PIR2); //receives the PIR2 values
  delay (200);
}

void ultrasonic(){
  // step 1: instruct sensor to read centimeters
  Wire.beginTransmission(112); // transmit to device #112 (0x70)
      // the address specified in the datasheet is 224 (0xE0)
      // but i2c adresssing uses the high 7 bits so it's 112
  Wire.write(byte(0x00)); //setsregister pointer to the command register (0x00)
  Wire.write(byte(0x51)); // command sensor to measure in "inches" (0x50)
      // use 0x51 for centimeters
      // use 0x52 for ping microseconds
  Wire.endTransmission(); // stop transmitting
  // step 2: wait for readings to happen
  delay(70); // datasheet suggests at least 65 milliseconds
  // step 3: instruct sensor to return a particular echo reading
  Wire.beginTransmission(112); // transmit to device #112
```

```
Wire.write(byte(0x02)); // sets register pointer to echo #1 register (0x02)
Wire.endTransmission(); // stop transmitting
// step 4: request reading from sensor
Wire.requestFrom(112, 2); // request 2 bytes from slave device #112
// step 5: receive reading from sensor
if(2 <= Wire.available()) // if two bytes were received
{
  oldreading = reading;
  reading = Wire.read(); // receive high byte (overwrites previous reading)
  reading = reading << 8; // shift high byte to be high 8 bits
  reading |= Wire.read(); // receive low byte as lower 8 bits
}
delay(25); // wait before next reading:
}

void presence(){
  //the follow condition memorizes if there was a motion during the 30seconds and print it
  int f = reading - oldreading;

  if((val1 == LOW) || (val2 == LOW) || (f > 15)){//pay attention to f value because of power
      //supply fluctuation effect in US distance reading
      //when is been used xbee(60 was the value selected)

    presence_state=1;
    digitalWrite(LED,HIGH);
  }
  else{
    digitalWrite(LED,LOW);
  }
}
```

```
void consumption(){
    for (int n=0; n < numberOfSamples; n++)
    {
        //Used for current offset removal
        lastSampleI = sampleI;
        lastFilteredI = filteredI;
        sampleI = analogRead(pinPot);
        //Apply digital high pass filters to remove 2.5V DC offset (centered on 0V)
        filteredI = 0.996*(lastFilteredI+sampleI-lastSampleI);
        //Root-mean-square method current
        //1) square current values
        sqI = filteredI * filteredI;
        //2) sum
        sumI += sqI;
    }
    //Calculated ratio constants
    float I_RATIO = CT_TURNS / CT_BURDEN_RESISTOR * 5 / 1024 * ICAL;
    Irms = I_RATIO*sqrt(sumI / numberOfSamples); //Calculation of the rms

    //condition to avoid when there's no current the code prints a current value due to resolution of ADC
    if(Irms < 0.3 || calibrationCT < 1 ){
        Irms=0.000;
        Serial.print(Irms,3);
        Serial.print(';');

        // Calculate power values
        apparentPower = Vrms * Irms;
        realPower = PF * apparentPower;
        Serial.print(apparentPower,1);
        Serial.print(';');
    }
```

```
Serial.print(realPower,1);
Serial.print(';');

// Calculate total kilowatt hours
last_kwhTime = kwhTime;
kwhTime = millis();
kilowattHour += (realPower / 1000) * ((kwhTime - last_kwhTime) / 3600000.0);
Serial.print(kilowattHour,3);
Serial.print(';');
// Reset sample totals
sumI = 0;
calibrationCT++;
}

else{
    Serial.print(Irms,3);
    Serial.print(';');

    // Calculate power values
    apparentPower = Vrms * Irms;
    realPower = PF * apparentPower;
    Serial.print(apparentPower,1);
    Serial.print(';');
    Serial.print(realPower,1);
    Serial.print(';');
    // Calculate total kilowatt hours
    last_kwhTime = kwhTime;
    kwhTime = millis();
    kilowattHour += (realPower / 1000) * ((kwhTime - last_kwhTime) / 3600000.0);
    Serial.print(kilowattHour,3);
}
```

```
Serial.print(';');
// Reset sample totals
sumI = 0;
calibrationCT++;
}
}

byte bcdToDec(byte val) {
// Convert binary coded decimal to normal decimal numbers
return ( (val/16*10) + (val%16) );
}

void runRTC(){
// Reset the register pointer
Wire.beginTransmission(DS1307_ADDRESS);
Wire.write(zero);
Wire.endTransmission();
Wire.requestFrom(DS1307_ADDRESS, 7);
second = bcdToDec(Wire.read());
minute = bcdToDec(Wire.read());
hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
day = bcdToDec(Wire.read());
month = bcdToDec(Wire.read());
year = bcdToDec(Wire.read());
}

void printDate(){
//print time data
Serial.print(day);
Serial.print(";");
```

```
Serial.print(month);
Serial.print(";");
Serial.print(year);
Serial.print(";");
Serial.print(hour);
Serial.print(";");
Serial.print(minute);
Serial.print(";");
Serial.print(second);
Serial.print(";");
}
```

Anexo C: Código de Programação do Processing

Anexo C.1: Classe Principal do Código

```
import napplet.*;
import processing.serial.*;
import cosm.*;

Serial port;
DataOut feed;

//Configuration to export data to COSM
String apiKey = "6e0hRzJ52eEY607-79zll4oYStGSAKwrRVJCaE10WURXcz0g";
String feedId = "70053";
PrintWriter dadosSRMEP; //entry to prepare file's creation
int count;

//variables to receive on serial port
float yeartime, monthtime, daytime;

float hourtime, minutetime, secondtime, oldminutetime ;
float presvalue, oldpresvalue;

float Irms;
```

```

float apparentPower,realPower;
float kilowattHour;
float timeX; //Calculates the total time, using the time function of processing
            // necessary to use on graphs Xaxis
            //Size of the window
int plotX=1150, plotY=800;
//Dimension of each graph window
int plotgW=575, plotgY=312;
//plotmw means the height of the main window
int plotmW=176;
color bgc=color(0, 153, 0); //background color of each window
color graphc=color(#4D4D4A); //graph line color
color textc=color(255); //text color on graph label
PFont plotFont; //font used in each graph

void setup() {
  size(plotX,plotY); //creates one window
  println(Serial.list()); // Print a list of the serial ports, for debugging purposes:
  port = new Serial(this,"COM14", 9600); // Open a serial port
  port.bufferUntil('\n'); // don't generate a serialEvent() unless you get a
                        // newline character:
  NAppletManager nappletManager = new NAppletManager(this);
  //define the initial points of each window created in our sketch
  nappletManager.createNApplet("Irms", 0, 0);
  nappletManager.createNApplet("RealPower", width/2, 0);
  nappletManager.createNApplet("kWh", 0, plotgY);
  nappletManager.createNApplet("Presence", width/2, plotgY);
  nappletManager.createNApplet("MainWindow", 0, height-plotmW);

  //font type and size to use

```

```

plotFont = createFont("SansSerif", 20);
textFont(plotFont);

//setup the text file to be created with description of several parameters
dadosSRMEP = createWriter("dadosSRMEP.csv");
dadosSRMEP.print("Date");
dadosSRMEP.print(";");
dadosSRMEP.print("Irms");
dadosSRMEP.print(";");
dadosSRMEP.print("Apparent_Power");
dadosSRMEP.print(";");
dadosSRMEP.print("Real_Power");
dadosSRMEP.print(";");
dadosSRMEP.print("kWh");
dadosSRMEP.print(";");
dadosSRMEP.println("Presence");

//connection to COSM feed
feed = new DataOut(this, apiKey, feedId); //intantiate feed
feed.setVerbose(false); //optional debug info
}

void draw() {
  background(0, 153, 0); //setup background color
  timeX=60.0*minutetime+secondtime+hourtime*3600.0; //Calculates the total time, using the time
                        //function of processing necessary to use on graphs Xaxis
  senddata(); //function which send data to Cosm

  if(minutetime != oldminutetime){ //just print if the time change using second from Arduino Time
                        //print the data received from Arduino on the text file

```

```
dadosSRMEP.print(nf(yeartime, 2,0)+"-"+nf(monthtime, 2,0)+"-"+nf(daytime, 2,0)+" "+nf(hourtime, 2,0)+":"+nf(minutetime, 2,0)+":"+nf(secondtime, 2,0));

dadosSRMEP.print(";");

dadosSRMEP.print(Irms);

dadosSRMEP.print(";");

dadosSRMEP.print(apparentPower);

dadosSRMEP.print(";");

dadosSRMEP.print(realPower);

dadosSRMEP.print(";");

dadosSRMEP.print(kilowattHour);

dadosSRMEP.print(";");

dadosSRMEP.println(presvalue);

dadosSRMEP.flush();

}

oldminutetime = minutetime;
}

void senddata() {
    feed.setStream(0, Irms); //send request (datastream id, new value)
    feed.setStream(1, apparentPower); //send request (datastream id, new value)
    feed.setStream(2, realPower); //send request (datastream id, new value)
    feed.setStream(3, kilowattHour); //send request (datastream id, new value)
    feed.setStream(4, presvalue); //send request (datastream id, new value)
}

void mousePressed() //necessary to know the coordinates of each point on the window
{
    println("Coordinates: " + mouseX + "," + mouseY);
}
```

```
void serialEvent(Serial port){
    String input = port.readStringUntil('\n');//read the information received on serial port until find a new //line

    if (input != null){ // if the string is not empty, do stuff with it:

        input = trim(input); // clean the empty spaces that aren't necessary of the string

        // The data is split into an array of Strings with a comma or asterisk as a delimiter and converted //into
        an array of integers.

        float [] infos =float (split(input, ",")); //fill a matrix with the imported variables from arduino

        if (infos.length >=11) //is the number of variables that are being imported from arduino
        {
            daytime = infos [0];
            monthtime = infos [1];
            yeartime = infos [2];
            hourtime = infos [3];
            minutetime = infos [4];
            secondtime = infos [5];

            Irms = infos [6];
            apparentPower = infos [7];
            realPower = infos [8];
            kilowattHour = infos [9];
            presvalue = infos [10]; //presence value
        }
    }
}
```

Anexo C.2: Classe para Amostragem dos Valores

```
public class MainWindow extends NApplet {

    //Load Images
```



```

PImage img1;
PImage img2;
PImage img3;
PImage img4;
PImage img5;
PImage img6;
PImage img7;
PFont font;

//int color1 = #516C7C;
//int color2 = 200;
int xe=75; //x ellipse
int ye=75; //y ellipse
int a=85; //y distance among ellipses
int b=370; //x distance among ellipses
//text positions
int xtext1= 110;int xtext2= xtext1+b;int xtext3=xtext2+b;
int x1=50;//x position of image ellipses
int y1=45;//y image ellipse 1
int x2=x1+b;//y image ellipse 2
int y2=y1+a;//y image ellipse 3
int x3=x2+b;//y iamge ellipse 4

void setup()
{
  size(plotX,plotgWY); //be aware about the size of the control window
  background(bgc);

  //Chama as imagens presentes na pasta data para o programa
  img1 = loadImage("icon60-current.png");
  img2 = loadImage("icon60-power.png");
  img3 = loadImage("icon60-energy.png");

```

```

img4 = loadImage("icon60-people.png");
img5 = loadImage("icon60-money.png");
img6 = loadImage("icon60-co2.png");
  font = loadFont( "Calibri-48.vlw" );
// font1 = loadFont("Tahoma-Bold-48.vlw");
  smooth();
}

void draw() {
  background(bgc);
  drawImages();
  drawValues();
}

void drawImages()
{
  //Draw images and ellipses
  fill(255);
  stroke(2);
  ellipse(x1,y1,x1,x2,y1,y2);
  image(img1, x1-30, y1-30);
  ellipse(x1,y2,x1,x2,y1,y2);
  image(img2, x1-28, y2-28);
  ellipse(x2,y1,x2,x3,y1,y2);
  image(img3, x2-27, y1-29);
  ellipse(x2,y2,x2,x3,y1,y2);
  image(img4, x2-29, y2-33);
  ellipse(x3,y1,x3,x4,y1,y2);
  image(img5, x3-30, y1-30);
  ellipse(x3,y2,x3,x4,y1,y2);
  image(img6, x3-28, y2-30);

```

```
stroke(2);
smooth();
}

void drawValues ()
{
//current values
textFont(font,60);
text(Irms,xtext1-10,y1+20);
//int power = int(apparentPower*1);
String power = nf(realPower, 4,0); //convert float apparent power received from Arduino in an
                                     // integer number with 4 digits
text(power,xtext1+10,y2+20);
text(kilowattHour,xtext2-10,y1+20);
float money = kilowattHour*0.08; //price calculation assuming 0,13€/kWh
text(money,xtext3-10,y1+20); //write the money
float co2 = kilowattHour*0.64; //co2 kg calculation assuming 0,64kgCo2/kWh electricity using
                                     // bibliography
text(co2,xtext3-10,y2+20); //write the co2
textSize(35);
if(presvalue == 1) {
    count = 1;
}
if(count > 0){
    count++;
    text("há\nmovimento",xtext2,y2-8);
    //delay(100);
}
if(count == 30){
    count = 0;
```

```
    }
    if (count == 0 && presvalue == 0) {
        text("não há\nmovimento",xtext2,y2-8);
        //delay(100);
    }
    //Units
    textSize (26);
    text("A",xtext1+184,y1+17);
    text("W",xtext1+184,y2+17);
    text("kWh",xtext2+184,y1+17);
    text("€",xtext3+184,y1+17);
    text("kg",xtext3+184,y2+17);
}
}
```

Anexo C.3: Classe do Gráfico da Corrente Eficaz

```
public class Irms extends NApplet {

    float dataMin, dataMax;
    float plotX1, plotY1;
    float plotX2, plotY2;
    float labelX, labelY;
    //Set the time data
    int [] time = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24};
    int timeInterval = 1;
    int timeMin = 0;
    int timeMax = 24;
    float volumeInterval = 2;    //scale of graph axis
    float volumeIntervalMinor = 2; //difference between axis
```

```
float lineX = 30;//To start next to the Y axis
```

```
PFont plotFont;
```

```
void setup ()
```

```
{
  size(plotwX,plotwY);
  background(bgc);
  plotX1 = lineX;
  plotX2 = width - 20;
  plotY1 = 30;
  plotY2 = height - 25;
  dataMin = 0;
  dataMax = 20;
  smooth();
}
```

```
void draw ()
```

```
{
  // Show the plot area as a white box
  rectMode(CORNERS);
  stroke(255);
  strokeWeight(1);
  rect(plotX1, plotY1, plotX2, plotY2);
  stroke(2);
  smooth();
  fill(bgc);
  noStroke();
  rect(0,0,width,plotY1);
  rect(0,0,plotX1,height);
  rect(0,height,width,plotY2);
```

```
rect(plotX2,0,width,height);
```

```
//draw two lines to erase small white parts above and on the left of each graph
```

```
stroke(bgc);
```

```
line(plotX1, plotY1, plotX2, plotY1);
```

```
line(plotX1, plotY1, plotX1, plotY2);
```

```
drawYearLabels();
```

```
drawVolumeLabels();
```

```
drawTitle();
```

```
drawDataLine();
```

```
smooth();
```

```
}
```

```
void drawTitle() { //draw the title of the graph
```

```
  fill(textc);
```

```
  textSize(20);
```

```
  textAlign(LEFT);
```

```
  String title = "Irms (A)";
```

```
  text(title, (width/2)-40, plotY1-3);
```

```
}
```

```
void drawYearLabels()
```

```
{
```

```
  fill(textc);
```

```
  textSize(10);
```

```
  textAlign(CENTER);
```

```
  // Use thin lines to draw the grid
```

```
  for (int row = 0; row < 25; row++) {
```

```
    if (time[row] % timeInterval == 0) {
```

```
      float x = map(time[row], timeMin, timeMax, plotX1, plotX2);
```

```
      text(time[row], x, plotY2 + textAscent() + 10);
```

```
stroke(graphic);
strokeWeight(0.00001);
line(x, plotY1, x, plotY2);
line(plotX1,plotY2,plotX2,plotY2);
line(plotX2,plotY1,plotX2,plotY2);
stroke(0);
strokeWeight(1);
line(x, plotY2+4, x, plotY2);
}
}
}
void drawVolumeLabels() {
  fill(textc);
  textSize(10);
  textAlign(RIGHT);
  stroke(0);
  strokeWeight(1);
  for (float v = dataMin; v <= dataMax; v += volumeIntervalMinor) {
    if (v % volumeIntervalMinor == 0) { // If a tick mark
      float y = map(v, dataMin, dataMax, plotY2, plotY1);
      if (v % volumeInterval == 0) { // If a major tick mark
        float textOffset = textAscent()/2; // Center vertically
        if (v == dataMin) {
          textOffset = 0; // Align by the bottom
        } else if (v == dataMax) {
          textOffset = textAscent(); // Align by the top
        }
        text(floor(v), plotX1 - 10, y + textOffset);
        line(plotX1 - 4, y, plotX1, y); // Draw major tick
        line(plotX1-2, y+13, plotX1, y+13); //line to draw middle lines
```

```
    } else {
      //line(plotX1 - 2, y, plotX1, y); // Draw minor tick
    }
  }
}
void drawDataLine() {
  stroke(graphic);
  strokeWeight(1);
  noFill();
  float maptime=map(timeX,0,86400,plotX1,plotX2); //map the respective value received from serial
                                                    // port according to x axis
  float mapIrms = map(Irms, dataMin, dataMax, plotY2, plotY1); //map the respective value received
                                                                // from serial port
  according to y axis
  line (maptime,plotY2,maptime,mapIrms);

  if (timeX==86340) //1 minute before midnight erase the graph
  {
    background (255);
  }
}
}
```

Anexo C.4: Classe do Gráfico da Potência Ativa

```

public class RealPower extends NApplet {

float dataMin, dataMax;
float plotX1, plotY1;
float plotX2, plotY2;
float labelX, labelY;
//Set the time data
int [] time = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24};
int timeInterval = 1;
int timeMin = 0;
int timeMax = 24;
float volumeInterval = 200;    //scale of graph axis
float volumeIntervalMinor = 200; //difference between axis
float lineX = 30;//To start next to the Y axis
PFont plotFont;

void setup ()
{
    size(plotgwX,plotgwY);
    background(bgc);
    plotX1 = lineX;
    plotX2 = width - 20;
    plotY1 = 30;
    plotY2 = height - 25;
    dataMin = 0;
    dataMax = 2000;
    smooth();
}

```

```

void draw ()
{
    // Show the plot area as a white box
    rectMode(CORNERS);
    stroke(255);
    strokeWeight(1);
    rect(plotX1, plotY1, plotX2, plotY2);
    stroke(2);
    smooth();
    fill(bgc);
    noStroke();
    rect(0,0,width,plotY1);
    rect(0,0,plotX1,height);
    rect(0,height,width,plotY2);
    rect(plotX2,0,width,height);
    //draw two lines to erase small white parts above and on the left of each graph
    stroke(bgc);
    line(plotX1, plotY1, plotX2, plotY1);
    line(plotX1, plotY1, plotX1, plotY2);
    drawYearLabels();
    drawVolumeLabels();
    drawTitle();
    drawClock();
    drawDataLine();
    smooth();
}

void drawClock(){ //draw a clock on this part of the window
    fill(textc);

```

```
    textSize(20);
    textAlign(LEFT);
    //text(nf(hourtime, 2) + ":" + nf(minutetime, 2) + ":" + nf(secondtime(), 2), (width)-100, plotY1-3);
    //text(inthourtime + ":" + intminutetime + ":" + intsecondtime, (width)-100, plotY1-3);
    text(nf(hourtime, 2,0) + ":" + nf(minutetime, 2,0) , (width)-70, plotY1-3);
}
```

```
void drawTitle() { //draw the title of the graph
    fill(textc);
    textSize(20);
    textAlign(LEFT);
    String title = "Power (W)";
    text(title, (width/2)-40, plotY1-3);
}
```

```
void drawYearLabels()
{
    fill(textc);
    textSize(10);
    textAlign(CENTER);
```

```
    // Use thin, blue lines to draw the grid
    for (int row = 0; row < 25; row++) {
        if (time[row] % timeInterval == 0) {
            float x = map(time[row], timeMin, timeMax, plotX1, plotX2);
            text(time[row], x, plotY2 + textAscent() + 10);
            stroke(graphc);
            strokeWeight(0.00001);
            line(x, plotY1, x, plotY2);
            line(plotX1, plotY2, plotX2, plotY2);
```

```
        line(plotX2, plotY1, plotX2, plotY2);
        stroke(0);
        strokeWeight(1);
        line(x, plotY2+4, x, plotY2);
    }
}
```

```
void drawVolumeLabels() {
    fill(textc);
    textSize(10);
    textAlign(RIGHT);
    stroke(0);
    strokeWeight(1);

    for (float v = dataMin; v <= dataMax; v += volumeIntervalMinor) {
        if (v % volumeIntervalMinor == 0) { // If a tick mark
            float y = map(v, dataMin, dataMax, plotY2, plotY1);
            if (v % volumeInterval == 0) { // If a major tick mark
                float textOffset = textAscent()/2; // Center vertically
                if (v == dataMin) {
                    textOffset = 0; // Align by the bottom
                } else if (v == dataMax) {
                    textOffset = textAscent(); // Align by the top
                }
                text(floor(v), plotX1 - 10, y + textOffset);
                line(plotX1 - 4, y, plotX1, y); // Draw major tick
                line(plotX1-2, y+13, plotX1, y+13); //line to draw midle lines
            } else {
                //line(plotX1 - 2, y, plotX1, y); // Draw minor tick
```

```

    }
  }
}

void drawDataLine() {
  stroke(graphic);
  strokeWeight(1);
  noFill();

  float maptime=map(timeX,0,86400,plotX1,plotX2); //map the respective //value
  received from serial port according to x axis

  float maprealPower = map(realPower, dataMin, dataMax, plotY2, plotY1);
    //map the respective value
    // received from serial port according to y axis
  line (maptime,plotY2,maptime,maprealPower);

  if (timeX==86340) //1 minute before midnight erase the graph
  {
    background (255);
  }
}
}

```

Anexo C.5: Classe do Gráfico da Energia Consumida

```

public class kWh extends NApplet {

  float dataMin, dataMax;
  float plotX1, plotY1;
  float plotX2, plotY2;
  float labelX, labelY;

```

```

//Set the time data
int [] time = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24};

int timeInterval = 1;
int timeMin = 0;
int timeMax = 24;

float volumeInterval = 5;    //diference between axis
float volumeIntervalMinor = 5; //scale of graph axis
float lineX = 30;//To start next to the Y axis
PFont plotFont;

void setup ()
{
  size(plotgwX,plotgwY);
  background(bgc);
  plotX1 = lineX;
  plotX2 = width - 20;
  plotY1 = 30;
  plotY2 = height - 25;
  dataMin = 0;
  dataMax = 50;
  smooth();
}

void draw ()
{
  // Show the plot area as a white box
  rectMode(CORNERS);
  stroke(255);
  strokeWeight(1);
  rect(plotX1, plotY1, plotX2, plotY2);

```

```
stroke(2);
smooth();
fill(bgc);
noStroke();
rect(0,0,width,plotY1);
rect(0,0,plotX1,height);
rect(0,height,width,plotY2);
rect(plotX2,0,width,height);
//draw two lines to erase small white parts above and on the left of each graph
stroke(bgc);
line(plotX1, plotY1, plotX2, plotY1);
line(plotX1, plotY1, plotX1, plotY2);
drawYearLabels();
drawVolumeLabels();
drawTitle();
drawDataLine();
smooth();
}

void drawTitle() { //draw the title of the graph
fill(textc);
textSize(20);
textAlign(LEFT);
String title = "Energy (kWh)";
text(title, (width/2)-40, plotY1-3);
}

void drawYearLabels()
{
fill(textc);
```

```
textSize(10);
textAlign(CENTER);

for (int row = 0; row < 25; row++) { // Use thin lines to draw the grid
if (time[row] % timeInterval == 0) {
float x = map(time[row], timeMin, timeMax, plotX1, plotX2);
text(time[row], x, plotY2 + textAscent() + 10);
stroke(graphc);
strokeWeight(0.00001);
line(x, plotY1, x, plotY2);
line(plotX1,plotY2,plotX2,plotY2);
line(plotX2,plotY1,plotX2,plotY2);
stroke(0);
strokeWeight(1);
line(x, plotY2+4, x, plotY2);
}
}
}

void drawVolumeLabels() {
fill(textc);
textSize(10);
textAlign(RIGHT);
stroke(0);
strokeWeight(1);

for (float v = dataMin; v <= dataMax; v += volumeIntervalMinor) {
if (v % volumeIntervalMinor == 0) { // If a tick mark
float y = map(v, dataMin, dataMax, plotY2, plotY1);
if (v % volumeInterval == 0) { // If a major tick mark
```



```

float textOffset = textAscent()/2; // Center vertically
if (v == dataMin) {
    textOffset = 0;          // Align by the bottom
} else if (v == dataMax) {
    textOffset = textAscent(); // Align by the top
}
text(floor(v), plotX1 - 10, y + textOffset);
line(plotX1 - 4, y, plotX1, y); // Draw major tick
line(plotX1-2, y+13, plotX1, y+13); //line to draw middle lines
} else {
    //line(plotX1 - 2, y, plotX1, y); // Draw minor tick
}
}
}
}

void drawDataLine() {
    stroke(graphc);
    strokeWeight(1);
    noFill();
    float maptime=map(timeX,0,86400,plotX1,plotX2); //map the respective value received from serial
                                                    //port according to x axis
    float mapkilowattHour = map(kilowattHour, dataMin, dataMax, plotY2, plotY1); //map the
                                                    //respective value received from serial port according to y axis

    line (maptime,plotY2,maptime,mapkilowattHour);
    if (timeX==86340) //1 minute before midnight erase the graph
    {
        background (255);
    }
}

```

```

}
}

```

Anexo C.6: Classe do Gráfico da Presença

```

public class Presence extends NApplet {

    float dataMin, dataMax;
    float plotX1, plotY1;
    float plotX2, plotY2;
    float labelX, labelY;

    //Set the time data
    int [] time = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24};
    int timeInterval = 1;
    int timeMin = 0;
    int timeMax = 24;
    float volumeInterval = 1; //scale of graph axis
    float volumeIntervalMinor = 0.5; //difference between axis
    float lineX = 30; //To start next to the Y axis
    PFont plotFont;

    void setup ()
    {
        size(plotgwX,plotgwY);
        background(bgc);
        plotX1 = lineX;
        plotX2 = width - 20;
        plotY1 = 30;
        plotY2 = height - 25;
        //labelX = 18;
        //labelY = height - 10;
    }
}

```

```
dataMin = 0;
dataMax = ceil( 1/ volumeInterval) * volumeInterval;
smooth();
}

void draw ()
{
    // Show the plot area as a white box
    rectMode(CORNERS);
    stroke(255);
    strokeWeight(1);
    rect(plotX1, plotY1, plotX2, plotY2);
    stroke(2);
    smooth();
    fill(bgc);
    noStroke();
    rect(0,0,width,plotY1);
    rect(0,0,plotX1,height);
    rect(0,height,width,plotY2);
    rect(plotX2,0,width,height);

    //draw two lines to erase small white parts above and on the left of each graph
    stroke(bgc);
    line(plotX1, plotY1, plotX2, plotY1);
    line(plotX1, plotY1, plotX1, plotY2);
    drawYearLabels();
    drawVolumeLabels();
    drawTitle();
    drawDataLine();
    smooth();
}
```

```
void drawTitle() { //draw the title of the graph
    fill(textc);
    textSize(20);
    textAlign(LEFT);
    String title = "Presence 0=No 1=Yes";
    text(title, (width/2)-100, plotY1-3);
}

void drawYearLabels()
{
    fill(textc);
    textSize(10);
    textAlign(CENTER);
    // Use thin, blue lines to draw the grid
    for (int row = 0; row < 25; row++) {
        if (time[row] % timeInterval == 0) {
            float x = map(time[row], timeMin, timeMax, plotX1, plotX2);
            text(time[row], x, plotY2 + textAscent() + 10);
            stroke(graphc);
            strokeWeight(0.00001);
            line(x, plotY1, x, plotY2);
            line(plotX1,plotY2,plotX2,plotY2);
            line(plotX2,plotY1,plotX2,plotY2);
            stroke(0);
            strokeWeight(1);
            line(x, plotY2+4, x, plotY2);
        }
    }
}
```

```
void drawVolumeLabels() {
  fill(textc);
  textSize(10);
  textAlign(RIGHT);
  stroke(0);
  strokeWeight(1);

  for (float v = dataMin; v <= dataMax; v += volumeIntervalMinor) {
    if (v % volumeIntervalMinor == 0) { // If a tick mark
      float y = map(v, dataMin, dataMax, plotY2, plotY1);
      if (v % volumeInterval == 0) { // If a major tick mark
        float textOffset = textAscent()/2; // Center vertically
        if (v == dataMin) {
          textOffset = 0; // Align by the bottom
        } else if (v == dataMax) {
          textOffset = textAscent(); // Align by the top
        }
        text(floor(v), plotX1 - 10, y + textOffset);
        line(plotX1 - 4, y, plotX1, y); // Draw major tick
        //line(plotX1-2, y+31, plotX1, y+31); //line to draw middle lines
      } else {
        //line(plotX1 - 2, y, plotX1, y); // Draw minor tick
      }
    }
  }
}
```

```
void drawDataLine() {
  stroke(graphc);
```

```
strokeWeight(1);
noFill();
float maptime=map(timeX,0,86400,plotX1,plotX2); //map the respective value received from serial
// port according to x axis

float mappresvalue = map(presvalue, 0, 1, plotY2, plotY1);
line (maptime,plotY2,maptime,mappresvalue);
lineX ++;

if (timeX==86340) //1 minute before midnight erase the graph
{
  background (255);
}
}
}
```